

University of Windsor

Scholarship at UWindsor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

10-7-2019

Green Supply Chain Network Design

Penelopi Tatiana Krikella

University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

Recommended Citation

Krikella, Penelopi Tatiana, "Green Supply Chain Network Design" (2019). *Electronic Theses and Dissertations*. 7816.

<https://scholar.uwindsor.ca/etd/7816>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

GREEN SUPPLY CHAIN NETWORK DESIGN

by

Penelopi Tatiana Krikella

A Thesis

Submitted to the Faculty of Graduate Studies
through the Department of Mathematics and Statistics
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada

2019

© 2019 Penelopi Tatiana Krikella

GREEN SUPPLY CHAIN NETWORK DESIGN
by

Penelopi Tatiana Krikella

APPROVED BY:

A. Alfakih
Department of Mathematics and Statistics

E. Selvarajah
Odette School of Business

R. Caron, Advisor
Department of Mathematics and Statistics

October 7th, 2019

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication. I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix. I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

This thesis presents an analysis of Elhedhli and Merrick's paper, *Green supply chain network design to reduce carbon emissions* [1]. Elhedhli and Merrick proposed a supply chain model to find the optimal placement of distribution centres (DCs) while minimizing transportation costs, the fixed cost of opening the DC and carbon emissions costs. They found that considering carbon emissions creates a pull to reduce the vehicle kilometers travelled, and results in a supply chain network with more DCs opened compared to a supply chain network that does not consider carbon emissions. This thesis is an exploration of the model and solution methods proposed by Elhedhli and Merrick. An overview of optimization topics, a description of Lagrangian relaxation and an outline of decomposition methods is included to provide background knowledge for the description of the green supply chain network model, and its solution methods. We use the algorithm suggested by Elhedhli and Merrick to find the optimal solution of three supply chain networks: one considering zero emissions costs, one considering moderate emissions costs and one considering high emissions costs. The results obtained from the three scenarios confirm the results of Elhedhli and Merrick.

Dedication

I would like to dedicate this Master's thesis to my grandfather, who passed away in the first year of my research.

Acknowledgements

I would like to thank my thesis advisor Dr. Richard Caron, for his valuable and constructive suggestions during the development of this thesis. He was always available when I ran into a road block, or had any questions regarding my research or writing. His jokes and positive presence helped my stress levels throughout this year as well.

I would also like to acknowledge the help and support from my boyfriend, Will. Thank you for the countless hours of teaching me how to code in MatLab, and for supporting and encouraging me through the process of researching and writing. I could not have done this without you.

Finally, I must express my very profound gratitude to my parents and my friends for supporting me, and providing me with happy distractions when my research became too hectic. You all kept me sane.

Thank you.

Table of Contents

Author's Declaration of Originality	iii
Abstract	iv
Dedication	v
Acknowledgements	vi
List of Figures	viii
Chapter 1. Introduction	1
1.1. Scope and Motivation of Research	1
1.2. Chapter Structure	3
1.3. Optimization	4
1.3.1. Linear Programming	4
1.3.2. Classic Optimization Problems	7
1.3.3. Convexity	13
1.4. Penalty Methods and Lagrangian Relaxation	17
1.5. Decomposition	21
Chapter 2. Green Supply Chain Network Design	28
2.1. Introduction	28
2.2. Elhedhli and Merrick's Model	29
2.3. Solution Strategy	31
2.4. Numerical Testing	42
Chapter 3. Results and Findings	46
3.1. Introduction	46
3.2. Results and Findings	47
Chapter 4. Conclusions	51
4.1. Conclusion	51
4.2. Summary of Contributions	51
4.3. Future Work	52
Bibliography	53
Appendix A	55
Problem Reformulations for CPLEX Compatibility	55
Vita Auctoris	60

List of Figures

1.1 A Transportation Problem	9
1.2 An Assignment Problem	10
1.3 A Facility Location Problem	12
1.4 The Lagrangian Function	20
1.5 An Optimization Problem with Complicating Constraints	21
2.1 Vehicle weight vs. CO ₂ emissions at 100km/h	43
3.1 Network Design with Zero Emissions Costs	48
3.2 Network Design with Moderate Emissions Costs	49
3.3 Network Design with High Emissions Costs	50

CHAPTER 1

Introduction

1.1. Scope and Motivation of Research

With the globalization of supply chains, the distances traveled has grown considerably, which in turn has increased the amount of daily vehicle emissions. As of 2019, transportation via combustion engine vehicles accounted for 28% of the Canadian greenhouse gas (CHG) inventory [2]. Integrating environmental costs into supply chains is now prevalent as we are realizing the effect carbon emissions have on the planet.

In their paper, Elhedhli and Merrick [1] develop a green supply chain design model which considers the cost of carbon emissions, in addition to fixed costs and transportation costs. Their proposed supply chain network is a three echelon model, which means it includes three levels: plants, DCs and customers. The goal of their supply chain network is to find the optimal locations of distribution centres (DCs), as well as the optimal assignments of plants to DCs and DCs to customers, which minimizes the overall cost of the network. Elhedhli and Merrick use published experimental data to derive a function that represents emissions costs. Their resulting model, a mixed integer programming problem, is the minimization of a concave function, which is a computationally difficult problem [3]. To solve the problem, the Lagrangian relaxation method is used which allows the problem to be decomposed into subproblems. The characteristics of the original problem are present in the subproblems, which results in achieving a strong Lagrangian bound. Elhedhli and Merrick then propose a primal heuristic to generate feasible solutions to their original problem. Their paper ends with numerical testing, in which they generate problems to test if their proposed method is effective in finding good solutions.

This thesis is an exploration of Elhedhli and Merrick's paper. The originality of this Master's thesis lies in the organization and presentation. We provide the details regarding the derivation of Elhedhli and Merrick's proposed problem, as well as the details to their solution strategy and algorithm. Chapter 1 is the introductory chapter to this thesis. It contains four sections, which cover the topics of linear programming, classic optimization problems, convexity, Lagrangian relaxation and decomposition. Chapter 2 is an analysis of Elhedhli and Merrick's paper. In this chapter, the derivation of the model is described, and the solution method of the model is outlined. The final section of this chapter describes the algorithm we use in our numerical testing, which is based on the algorithm proposed by Elhedhli and Merrick. Chapter 3 displays the results and findings of our numerical testing. We show that the results we obtain coincide with the results of Elhedhli and Merrick.

Chapter 4 is the concluding chapter of this thesis. In this chapter, we give a summary of our findings and our contributions. We also present ideas that could be implemented into the green supply chain model as future work.

1.2. Chapter Structure

In this chapter, we introduce multiple concepts over three sections that are referenced in Elhedhli and Merrick's problem formulation and solution strategy.

Section 1.3 is an overview of some concepts in optimization. It consists of three subsections. Subsection 1.3.1 covers linear programming. In this section, we discuss basic notation and definitions, the Simplex method, duality theory and the necessary and sufficient conditions of optimality. Subsection 1.3.2 provides some details on three classic optimization problems: the transportation problem, the assignment problem, and the facility location problem. Each description includes a general problem, an example, a graph and a solution algorithm. Subsection 1.3.3 presents definitions and theorems related to convexity.

Section 1.4 provides information that may help in understanding the solution strategy of Elhedhli and Merrick. In their paper, Elhedhli and Merrick use Lagrangian relaxation as a tool to solve their model. This section includes the theory behind the exterior penalty method, the exact penalty method, as well as the Lagrangian relaxation technique.

Section 1.5 covers definitions related to decomposition, as well as outlines a decomposition algorithm. At the end of the section, there is a comparison of a well-known decomposition algorithm to the Lagrangian relaxation algorithm.

1.3. Optimization

1.3.1. Linear Programming. The source of the material in this section is Best and Ritter's [4].

Consider the general linear programming problem (LP), which is to find the values $x \in \mathbb{R}^n$ that will

$$(LP) \quad \min \quad z = c^\top x \quad (1)$$

$$\text{s.t.} \quad Ax = b, \quad (2)$$

$$x \geq 0. \quad (3)$$

We refer to (1) as the objective function. This is the function we are trying to minimize the value of, with respect to x . To solve the problem, x must satisfy the constraints represented by (2) and (3).

DEFINITION 1.1. At a given point \hat{x} , a constraint can either be violated, satisfied and inactive or satisfied and active.

- (1) The constraint $a_i^\top x \leq b_i$ is violated at \hat{x} if $a_i^\top \hat{x} > b_i$.
- (2) The constraint $a_i^\top x \leq b_i$ is satisfied and inactive at \hat{x} if $a_i^\top \hat{x} < b_i$.
- (3) The constraint $a_i^\top x \leq b_i$ is satisfied and active at \hat{x} if $a_i^\top \hat{x} = b_i$.
- (4) The constraint $a_i^\top x = b_i$ is violated at \hat{x} if $a_i^\top \hat{x} \neq b_i$.
- (5) The constraint $a_i^\top x = b_i$ is satisfied and active at \hat{x} if $a_i^\top \hat{x} = b_i$.

We denote the feasible region as $\mathcal{R} = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$. If $x \in \mathcal{R}$, we say x is a feasible solution. An optimal solution is a feasible solution x^* such that $c^\top x^* \leq c^\top x, \forall x \in \mathcal{R}$. When solving an LP, we are looking for the optimal solution.

In a linear program, there may exist redundancy in terms of the constraints. A constraint $g_k(x) \leq 0$ is redundant in the feasible region $\mathcal{R} = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0, \forall i\}$ if $\{x \in \mathbb{R}^n \mid g_i(x) \leq 0, \forall i\} = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0, \forall i \neq k\}$. In other words, a constraint is redundant if it is implied by other constraints. By including redundant constraint in a linear program, the computational effort of solving the program is affected, as time is wasted. We try to eliminate redundant constraints by removing them from the linear programming problem.

Each constraint of (LP) has a corresponding dual variable. We will denote the vector of dual variables associated with (2) and (3) by u , and v , respectively. We define v , the vector of dual variables associated with the non-negativity constraints, to be the reduced cost.

The gradient of a function is the vector of first partial derivatives. We will refer to a_i as the gradient of the i^{th} constraint. The gradient of the objective function is c .

LEMMA 1.2. *The negative gradient of the objective function, $-c$, points in the direction of maximal local decrease of the objective function $z = c^\top x$.*

DEFINITION 1.3 (Extreme Point). The point $\hat{x} \in \mathcal{R}$ is an extreme point of \mathcal{R} if the set of all gradient vectors of the constraints active at \hat{x} span \mathbb{R}^n

A result used in LP solution algorithms is if there is an optimal solution, then there exists an extreme point that is an optimal solution. This result is used in the Simplex algorithm, which is a popular solution algorithm for an LP. To use the Simplex algorithm, the LP must be in the same form as (1)-(3). This is called the Standard Simplex Form. The Simplex algorithm has two phases. In Phase *I*, we consider a "relaxed" problem in which artificial variables are added to each of the constraints. An artificial variable is one that is required to be 0, but in Phase *I* this requirement is relaxed, and the artificial variables are greater than or equal to 0. In this phase, there are two possible results: either all of the artificial variables are eliminated, or we cannot eliminate all of them. If they are all eliminated, then an extreme point is found and Phase *II* begins with that extreme point as a starting point. This phase also has two possible results: either an optimal solution is found, or we find that the objective function is unbounded from below.

An important topic in linear programming is duality theory. The Standard Simplex linear program given in (1)-(3) is called the primal. We define the dual of (1)-(3) to be the LP

$$(DLP) \quad \max \quad -b^\top u \quad (4)$$

$$\text{s.t.} \quad A^\top u \geq -c \quad (5)$$

Together, the two LPs are the primal-dual pair. Every linear programming problem has a corresponding dual problem. A relationship between the primal-dual pair is described in Theorem 1.4.

THEOREM 1.4. *The dual of the dual is the primal.*

PROOF. The dual of (1)-(3) is the LP

$$\begin{aligned} \max \quad & -b^\top u \\ \text{s.t.} \quad & A^\top u \geq -c \end{aligned}$$

which is equivalent to

$$\begin{aligned} \min \quad & b^\top u \\ \text{s.t.} \quad & -A^\top u \leq c. \end{aligned}$$

Now, the dual of the dual is

$$\begin{aligned} \max \quad & -c^\top y \\ \text{s.t.} \quad & (-A^\top)^\top y = -b, \\ & y \geq 0. \end{aligned}$$

Let $y = x$, and substitute into the above LP to get

$$\begin{aligned} \max \quad & -c^\top x \\ \text{s.t.} \quad & -Ax = -b, \\ & x \geq 0 \end{aligned}$$

which simplifies to

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0. \end{aligned}$$

□

There is also a relationship between the objective functions of the primal and its dual.

THEOREM 1.5 (Weak Duality). *If \hat{x} is primal feasible and \hat{u} is dual feasible, then $c^\top \hat{x} \geq -b^\top \hat{u}$.*

PROOF. We use the primal (1)-(3) and its dual (4)-(5). From dual feasibility we have

$$-A^\top \hat{u} \leq c. \tag{6}$$

and from primal feasibility we have $\hat{x} \geq 0$. We rewrite (6) as

$$c^\top \geq -u^\top A$$

and post multiply by \hat{x} to get

$$c^\top \hat{x} \geq -u^\top A \hat{x}. \tag{7}$$

From primal feasibility we have

$$A \hat{x} = b. \tag{8}$$

We pre-multiply (8) by $-\hat{u}^\top$ to get

$$-\hat{u}^\top A \hat{x} = -\hat{u}^\top b. \tag{9}$$

Combining (7) and (9) we get the result

$$c^\top \hat{x} \geq -b^\top \hat{u}.$$

□

The final topic in this subsection is the necessary and sufficient conditions for optimality. These conditions are also known as the Karush-Kuhn Tucker conditions (KKTCS).

THEOREM 1.6. *Consider the LP $\min\{c^\top x \mid Ax = b, x \geq 0\}$. The point x^* is an optimal solution to this LP if and only if there exists a vector u^* that, together with x^* satisfy*

- (1) $Ax = b, x \geq 0$ (Primal Feasibility)
- (2) $A^\top u - v = -c, v \leq 0$ (Dual Feasibility)
- (3) $v^\top x = 0$ (Complementary Slackness)

We proof the sufficiency of these conditions.

PROOF. Suppose there are vectors x^*, u^* and v^* which satisfy the optimality conditions in Theorem 1.6. We will show that x^* is optimal for $\min\{c^\top x \mid Ax = b, x \geq 0\}$. Since $Ax^* \leq b, x^* \geq 0$, we know that x^* is feasible. Let \hat{x} be any other feasible point. We will show that $c^\top x^* \leq c^\top \hat{x}$, i.e., that $c^\top(x^* - \hat{x}) \leq 0$.

From dual feasibility, we have that

$$\begin{aligned} A^\top u^* - v^* &= -c, \\ \implies -c^\top &= u^{*\top} A^\top - v^{*\top}. \end{aligned}$$

Consider

$$\begin{aligned} -c^\top(x^* - \hat{x}) &= (u^{*\top} A^\top - v^{*\top})(x^* - \hat{x}), \\ &= (u^{*\top} Ax^* - v^{*\top} x^*) - (u^{*\top} A\hat{x} - v^{*\top} \hat{x}), \\ &= u^{*\top} Ax^* - u^{*\top} A\hat{x} - v^{*\top} \hat{x}, && \text{by Complementary Slackness,} \\ &= u^{*\top}(Ax^* - A\hat{x}) - v^{*\top} \hat{x}, \\ &= u^{*\top}(b - b) - v^{*\top} \hat{x}, && \text{by Primal Feasibility,} \\ &= -v^{*\top} \hat{x}. \end{aligned}$$

We have

$$v^* \leq 0, \hat{x} \geq 0$$

from dual feasibility, and the feasibility of \hat{x} , respectively.

So, we conclude that

$$\begin{aligned} c^\top(x^* - \hat{x}) &= v^{*\top} \hat{x}, \\ &\leq 0. \end{aligned}$$

□

1.3.2. Classic Optimization Problems. The source for the material in this section is Ahuja et al.'s [11].

In this section, we introduce classic optimization problems that arise in this thesis. A general model, an example and a graph corresponding to the example is provided for each type of problem. Further, a brief overview of a solution algorithm of each problem is discussed.

DEFINITION 1.7. A graph is an ordered pair, $G = (N, A)$, where N is a set of nodes (also called vertices) and A is a set of arcs. A directed graph is a graph in which the arcs begin at a location, i , and end at a location, j . We denote this directed arc as (i, j) . A directed graph can also be called a directed network.

Throughout this chapter, the graphs/networks we discuss are all directed graphs/networks.

A transportation problem is a directed network where the node set N is partitioned into two subsets N_1 and N_2 , not necessarily equal in size. Each node in N_1 is a supply node, each node in N_2 is a demand node and for each arc (i, j) in the network, $i \in N_1$ and $j \in N_2$. The variable x_{ij} represents the number of units flowing over arc (i, j) . Further, each arc (i, j) has an associated per unit flow cost, c_{ij} .

A general transportation model is to

$$\begin{aligned}
 (TM) \quad & \min \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \\
 \text{s.t.} \quad & \sum_{j=1}^m x_{ij} = a_i, \quad \forall i = 1, \dots, n \\
 & \sum_{i=1}^n x_{ij} = b_j, \quad \forall j = 1, \dots, m \\
 & l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \\
 & x_{ij} \in \mathbb{Z}, \quad \forall (i, j)
 \end{aligned}$$

The transportation problem, (TM), is a balanced transportation problem. A transportation model is balanced if

$$\sum_{i=1}^n a_i = \sum_{j=1}^m b_j.$$

PROPERTY 1.8. A transportation problem will have feasible solutions if and only if the problem is balanced (Hillier and Lieberman, [6]).

Example 1.9 shows a balanced transportation problem, along with its graph represented by Figure 1.1 and its solution.

EXAMPLE 1.9.

$$\begin{array}{rcll}
 \min & 15x_{11} & + & 10x_{12} & + & 11x_{13} & + & 17x_{21} & + & 8x_{22} & + & 15x_{23} \\
 \text{s.t.} & x_{11} & + & x_{12} & + & x_{13} & & & & & & & = & 100 \\
 & & & & & & & x_{21} & + & x_{22} & + & x_{23} & = & 200 \\
 & x_{11} & + & & & & & x_{21} & & & & & = & 75 \\
 & & & x_{12} & + & & & & & x_{22} & & & = & 175 \\
 & & & & & x_{13} & + & & & & & x_{23} & = & 50 \\
 & x_{ij} & \geq & 0 & i = 1, 2, j = 1, 2, 3 & & & & & & & & .
 \end{array}$$

The nodes on the left, P_1 and P_2 , are the supply nodes. The nodes on the right, F_1 , F_2 and F_3 , are the demand nodes. The capacities of both supply and demand nodes are displayed outside of the nodes. The unit cost associated with each arc (i, j) is displayed on the left end of each arc. The numbers on the right end of each arc, in the black boxes, indicate how much product is being shipped on that

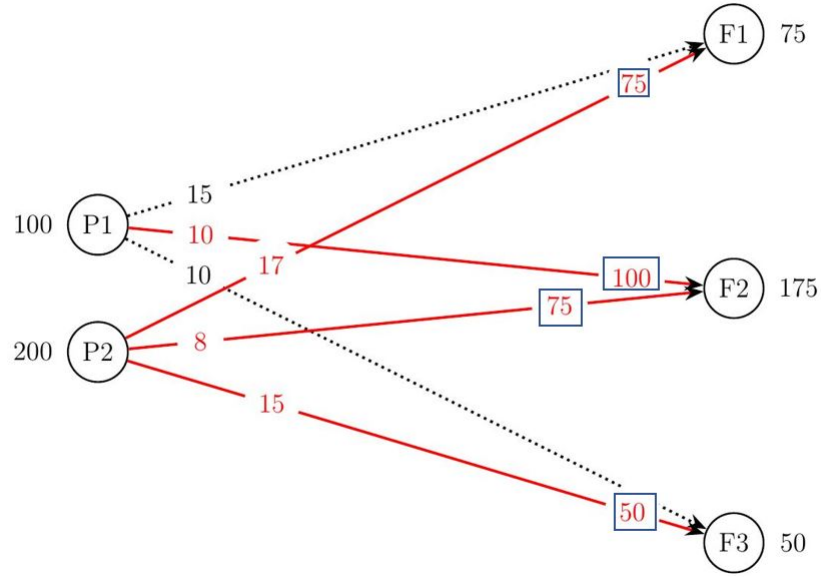


FIGURE 1.1. A Transportation Problem

arc. If an arc (i, j) is represented by a dotted line, then there is no product being shipped on that arc.

The transportation problem can be solved using the transportation simplex method. This method avoids the use of artificial variables, which in turn avoids the $(n+m-1)$ iterations to eliminate them.

An assignment problem is a special case of the transportation problem in which $a_i = b_j = 1, \forall i, j$ and $x_{ij} \in \{0, 1\}, \forall i, j$. A general model of an assignment problem is

$$\begin{aligned}
 (AM) \min \quad & \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \\
 \text{s.t.} \quad & \sum_{j=1}^m x_{ij} = 1, \quad \forall i = 1, \dots, n \\
 & \sum_{i=1}^n x_{ij} = 1, \quad \forall j = 1, \dots, m \\
 & x_{ij} \in \{0, 1\} \quad \forall i, j.
 \end{aligned}$$

The constraints of the assignment problem ensure that each supply node, i , is assigned to only one demand node, j . Also, each demand node j is only assigned to one supply node, i . Finally, in an assignment problem, the variables are binary. The variable, x_{ij} is 1 if node i is assigned to node j , and 0 otherwise.

Example 1.10 shows an assignment problem is shown, along with its graph represented by Figure 1.2 and its solution.

EXAMPLE 1.10.

$$\begin{array}{llllllllll}
\min & 15x_{11}+ & 30x_{12}+ & 10x_{13}+ & 17x_{21}+ & 17x_{22}+ & 20x_{23}+ & 17x_{31}+ & 30x_{32}+ & 25x_{33} \\
\text{s.t.} & x_{11}+ & x_{12}+ & x_{13} & & & & & & \\
& & & & x_{21}+ & x_{22}+ & x_{23} & & & \\
& & & & & & & x_{31}+ & x_{32}+ & x_{33} \\
& x_{11}+ & & & x_{21}+ & & & x_{31} & & \\
& & x_{12}+ & & & x_{22}+ & & & x_{32} & \\
& & & x_{13}+ & & & x_{23}+ & & & x_{33} \\
& x_{ij} & \in & \{0,1\} & \forall i,j. & & & & &
\end{array}
\begin{array}{l}
= 1 \\
= 1 \\
= 1 \\
= 1 \\
= 1 \\
= 1
\end{array}$$

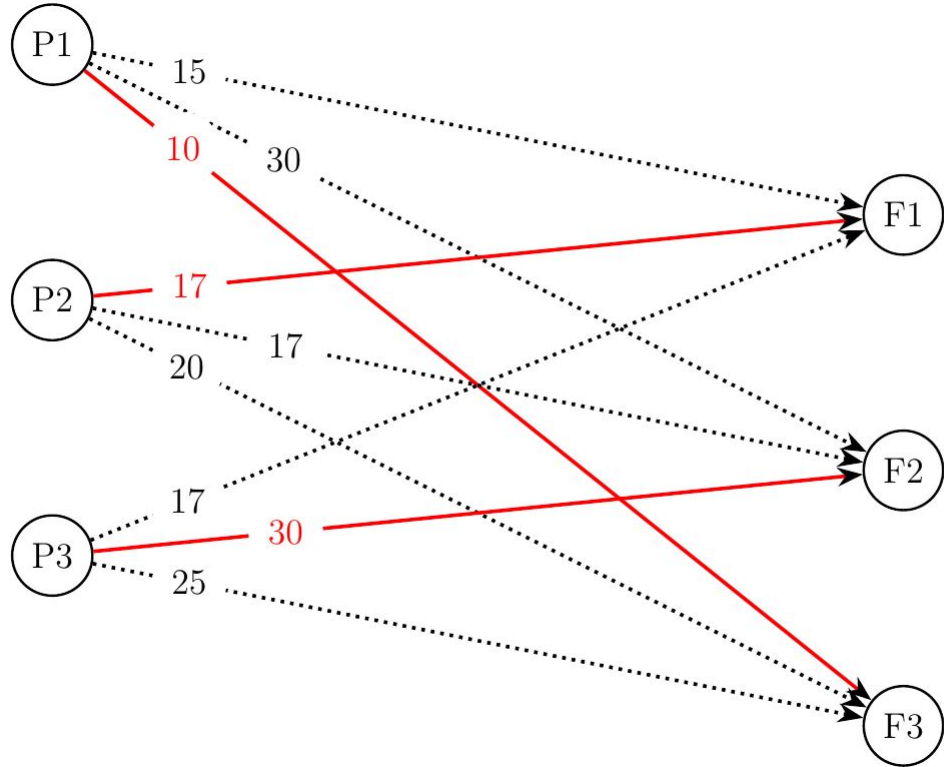


FIGURE 1.2. An Assignment Problem

If the arc, (i, j) , is represented by a dotted line, then the P_i is not assigned to F_j . The values on each arc (i, j) is the unit cost associated with that arc.

A well known algorithm to solve assignment problems is the Hungarian algorithm. This algorithm terminates within n_1 iterations, where n_1 is the number of supply nodes. In the example above, the algorithm would terminate within 3 iterations.

A facility location problem is a problem whose solution, if it exists, is the optimal placement of facilities to minimize transportation costs while considering other factors. Other factors could include fixed costs of opening a facility, risk, or in the case of this thesis, emissions costs. The problem is capacitated when each facility has a maximum capacity. In their paper, Elhedhli and Merrick consider a

capacitated facility location problem.

Consider n facilities and m customers. Let h_{jk} be the transportation cost from facility i , $i = 1, \dots, n$ to customer j , $j = 1, \dots, m$. Let g_i denote the fixed cost of opening facility i . Let v_j be the capacity for facility j . Let d_k be the demand of customer k . Let z_j be the binary variable where $z_j = 1$ if facility j is open, and 0 otherwise. Let x_{jk} be the number units shipped from facility j to customer k . These variables and indices are consistent with those in the model proposed by Elhedhli and Merrick, which is described in Chapter 2.

Then, the capacitated facility location problem is given by

$$(FLP) \quad \min \quad \sum_{j=1}^n \sum_{k=1}^m h_{jk} x_{jk} + \sum_{i=1}^n g_i z_i \quad (10)$$

$$\text{s.t.} \quad \sum_{j=1}^n x_{jk} = d_k \quad \forall k \quad (11)$$

$$\sum_{k=1}^m x_{jk} \leq v_j z_j \quad \forall j \quad (12)$$

$$z_j \in \{0, 1\} \quad \forall j \quad (13)$$

$$x_{jk} \in \mathbb{Z}_{\geq 0} \quad \forall j, k. \quad (14)$$

The symbol $\mathbb{Z}_{\geq 0}$ represents the space of non-negative integers. Constraint (11) ensures customer demand is satisfied. Constraint (12) ensures that the units being shipped from a facility j never exceeds the capacity of the facility j . Constraints (13) and (14) define z_j to be a binary variable and x_{jk} to be a non-negative integer, respectively. If each customer is sourced by one facility, we say that the above problem has single sourcing.

Example 1.11 shows a capacitated facility location problem, along with its graph represented by Figure 1.3 and its solution.

EXAMPLE 1.11. Consider a system with three distribution centres (DCs) and five customers. We wish to determine which DCs should open, as well as how much product is going from each DC to each customer to satisfy the demand of each customer.

Let $j = 1, 2, 3$ index the DCs, and let $k = 1, \dots, 5$ index the customers.

Let

$$[h_{jk}] = \begin{bmatrix} 50 & 75 & 100 & 100 & 30 \\ 70 & 70 & 85 & 100 & 50 \\ 55 & 60 & 60 & 120 & 45 \end{bmatrix}, [g_j] = \begin{bmatrix} 120 \\ 100 \\ 100 \end{bmatrix}, [v_j] = \begin{bmatrix} 250 \\ 200 \\ 200 \end{bmatrix}, [d_k] = \begin{bmatrix} 50 \\ 30 \\ 100 \\ 70 \\ 140 \end{bmatrix}$$

Then, we want to

$$\begin{aligned}
\min \quad & \sum_{j=1}^3 \sum_{k=1}^5 h_{jk} x_{jk} + \sum_{j=1}^3 g_j z_j \\
\text{s.t.} \quad & \sum_{j=1}^3 x_{jk} = d_k \quad \forall k \\
& \sum_{k=1}^5 x_{jk} \leq v_j z_j \quad \forall j \\
& z_j \in \{0, 1\} \quad \forall j \\
& x_{jk} \geq 0 \quad \forall j, k.
\end{aligned}$$

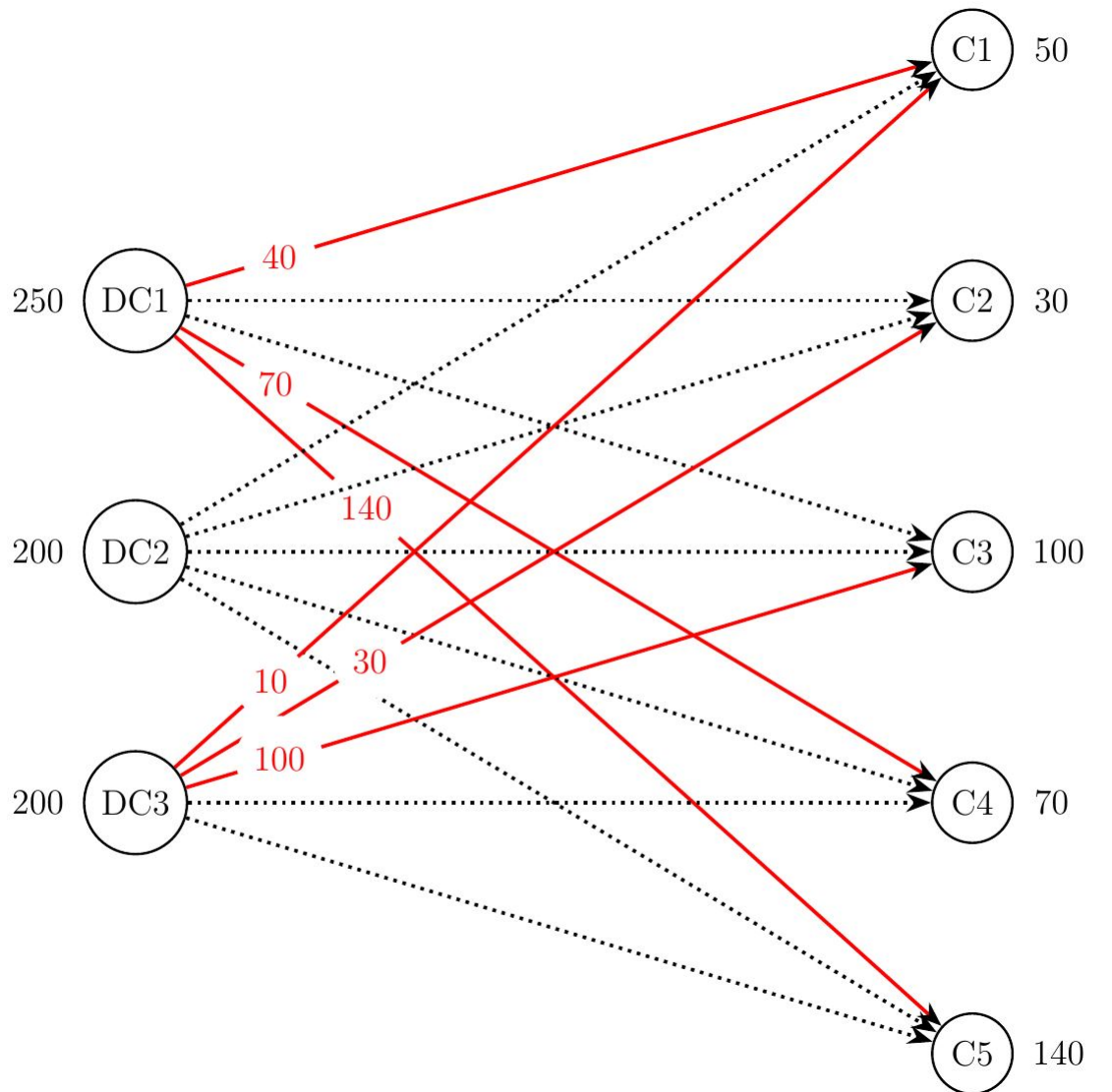


FIGURE 1.3. A Facility Location Problem

If an arc is represented by a dotted line, then there is no product being shipped on that arc. The numbers on the solid lines indicate how much product is being

shipped on that arc. We do not include the unit costs of shipping on an arc in the graph as to avoid clutter. From the solution, we see that DC2 is not open.

As the capacity facility location problem is an integer programming problem, algorithms used to solve integer programming problems can be used. An example of such an algorithm is the Branch and Bound method. This procedure involves searching for an optimal solution by partitioning the feasible region into subsets (branching), then pruning the enumeration by bounding the objective function values of the subproblems generated [7].

1.3.3. Convexity. The source for the material in this section is Rockafellar's [8].

We begin with the definitions of a convex set and a convex combination.

DEFINITION 1.12. A subset C of \mathbb{R}^n is convex if $(1 - \lambda)x + \lambda y \in C$ whenever $x \in C$, $y \in C$ and $0 < \lambda < 1$.

A convex set C is closed if it contains all of its limit points.

EXAMPLE 1.13. Consider the Standard Simplex linear programming problem given in (1)-(3). We will show that the feasible region $\mathcal{R} = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ is a convex set.

Let $x_1, x_2 \in \mathcal{R}$ and let $\lambda \in [0, 1]$. Since $Ax_1 = b$ and $Ax_2 = b$ then

$$\begin{aligned} A((1 - \lambda)x_1 + \lambda x_2) &= A(1 - \lambda)x_1 + A(\lambda x_2) \\ &= (1 - \lambda)Ax_1 + \lambda Ax_2 \\ &= (1 - \lambda)b + \lambda b \\ &= b. \end{aligned}$$

Since $x_1 \geq 0$ and $x_2 \geq 0$, we have

$$(1 - \lambda)x_1 + \lambda x_2 \geq 0$$

Since $(1 - \lambda)x_1 + \lambda x_2 \in \mathcal{R}$, \mathcal{R} is convex.

DEFINITION 1.14. A convex combination of vectors y_1, \dots, y_n is their linear combination

$$y = \sum_{i=1}^n \lambda_i y_i$$

with non-negative coefficients with unit sum, that is,

$$\lambda_i \geq 0, \quad \sum_{i=1}^n \lambda_i = 1$$

A line is uniquely determined by a point and a vector. For example,

$$x + \lambda s, \lambda \in \mathbb{R}$$

represents a line, from point x , extending in directions $\pm s$. A line extends in both directions infinitely. A half line, or a ray, is a line which extends in only one direction. For example,

$$x + \lambda s, \lambda \geq 0$$

is a ray. A line segment can be expressed by the convex combination of any two points in \mathbb{R}^n . For example,

$$(1 - \lambda)x_1 + \lambda x_2, 0 \leq \lambda \leq 1$$

is the line segment connecting x_1 and x_2 .

To conclude our discussion of convex sets, we introduce some topological definitions that will be used in Chapter 2 of this thesis.

DEFINITION 1.15. Let C be a non-empty convex set in \mathbb{R}^n . We say that C recedes in the direction $y \neq 0$ if and only if $x + \lambda y \in C$ for every $\lambda \geq 0$ and $x \in C$. The set of all vectors y satisfying these conditions, including $y = 0$, is called the recession cone of C and is denoted by 0^+C .

DEFINITION 1.16. Let C be a non-empty convex set. The set $(-0^+C) \cap 0^+C$ is called the lineality space of C . It consists of the zero vector and all the non-zero vectors y such that, for every $x \in C$, the line through x in the direction of y is contained in C .

We begin our overview of convex functions with the following definitions:

DEFINITION 1.17. Let f be a function whose values are real or $\pm\infty$ and whose domain is a subset S of \mathbb{R}^n . The epigraph of f , denoted by $\text{epi}f$, is the set

$$\{(x, \mu) | x \in S, \mu \in \mathbb{R}, \mu \geq f(x)\}.$$

We define f to be a convex function on S if the $\text{epi}f$ is a convex set. A concave function on S is a function whose negative is convex.

DEFINITION 1.18. The effective domain of a convex function f on S , denoted by $\text{dom}f$, is the set

$$\{x | f(x) < +\infty\}.$$

The dimension of $\text{dom}f$ is called the dimension of f .

We introduce three equivalent definitions of convex functions.

DEFINITION 1.19. Let C be a non-empty convex set. The function $f : C \rightarrow \mathbb{R}$ is convex if and only if

$$f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y) \quad 0 \leq \lambda \leq 1,$$

for every x and y in C .

DEFINITION 1.20. The function $f : C \rightarrow \mathbb{R}$ is convex if and only if

$$f((1 - \lambda)x + \lambda y) \geq f(x) + \lambda(\nabla_x f(x)(y - x))$$

whenever $\lambda \geq 0$ and $x, y \in C$.

DEFINITION 1.21. Let f be a twice differentiable real-valued function on an open convex set C in \mathbb{R}^n . Then f is convex on C if and only if its Hessian matrix is positive semi-definite for every $x \in C$.

Examples 1.22 and 1.23 show how we can prove functions are concave and/or convex.

EXAMPLE 1.22. To show that $f(x) = a \ln(x) + b$, $a > 0$, is concave on its domain, $(0, \infty)$, we show that it's negative is convex using Definitions 1.21. Since we are in \mathbb{R}^1 , we do not need to find the Hessian, we can simply take the second derivative of the function. If the value of the second derivative is negative for all values of x , then the function $f(x)$ is concave on its domain.

$$\begin{aligned} \frac{df(x)}{dx} &= \frac{a}{x} \\ \frac{d^2 f(x)}{dx^2} &= \frac{-a}{x^2} \end{aligned}$$

The negative of the second derivative will always be positive for values of x greater than 0, thus showing that $-f(x)$ is convex on its domain. Therefore, $f(x)$ is concave on its domain.

EXAMPLE 1.23. From Definition 1.21, we know that if the second derivative of a function, $f(x)$, is equal to 0 for all values of x , then $f(x)$ is both concave and convex. We will show that the linear function $f(x) = mx + b$ is both concave and convex.

$$\begin{aligned} \frac{df(x)}{dx} &= m \\ \frac{d^2 f(x)}{dx^2} &= 0 \end{aligned}$$

A well known property of concave functions is that a global minimum is achieved at some extreme point of the feasible domain. This property is a corollary to Theorem 1.24 below. Note that both the Theorem 1.24 and Corollary 1.25 involve finding a maximum solution of a convex function, which is equivalent to finding a minimum solution of a concave function.

THEOREM 1.24. *Let f be a convex function and let \mathbf{C} be a closed, convex set contained in the $\text{dom} f$. Suppose there are no half lines in \mathbf{C} on which f is unbounded above. Then*

$$\sup\{f(x)|x \in \mathbf{C}\} = \sup\{f(x)|x \in \mathbf{E}\}$$

where \mathbf{E} is the subset of \mathbf{C} consisting of the extreme points of $\mathbf{C} \cap \mathbf{L}^\perp$, \mathbf{L} being the lineality space of \mathbf{C} . The supremum relative to \mathbf{C} is attained only when the supremum relative to \mathbf{E} is attained.

The proof of this theorem is omitted as we are more interested in its corollary.

COROLLARY 1.25. *Let f be a convex function and let \mathbf{C} be a closed convex set contained in $\text{dom} f$. Suppose that \mathbf{C} contains no lines. Then, if the supremum of f relative to \mathbf{C} is attained at all, it is attained at some extreme point of \mathbf{C} .*

PROOF. If \mathbf{C} contains no lines, then $\mathbf{L}=\{0\}$ and $\mathbf{C} \cap \mathbf{L}^\perp = \mathbf{C}$. By Theorem 1.24, \mathbf{E} is the set containing extreme points of $\mathbf{C} \cap \mathbf{L}^\perp = \mathbf{C}$. Thus, \mathbf{E} is the set containing extreme points of \mathbf{C} . Further, the supremum relative to \mathbf{C} is attained only when supremum relative to \mathbf{E} is attained. As follows, the supremum of f relative to \mathbf{C} , if attained at all, is attained at some extreme point of \mathbf{C} . \square

1.4. Penalty Methods and Lagrangian Relaxation

The source for the material on penalty methods is Avriel's [9].

Suppose we want the minimum of a real-valued, continuous function f , defined on \mathbb{R}^n , on a proper subset X of \mathbb{R}^n .

Define

$$P(x) = \begin{cases} 0 & x \in X \\ +\infty & \text{otherwise.} \end{cases}$$

$P(x)$ is called the penalty function, as it imposes an infinite penalty on points outside of the feasible set. Consider the unconstrained minimization of the augmented objective function, F , given by

$$\min F(x) = f(x) + P(x).$$

A point x^* minimizes F if and only if it minimizes f over X . Solving the unconstrained minimization of F cannot be carried out in practice because of the infinite penalty on values outside of X . Instead, penalty methods consist of solving a sequence of unconstrained minimizations in which a penalty parameter is adjusted from one minimization to another, so that the sequence converges to an optimal point of the constrained problem. There are many different penalty methods, but we will discuss the exterior and the exact penalty method.

Consider the general non-linear programming problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \geq 0 \quad i = 1, \dots, m \\ & h_j(x) = 0 \quad j = 1, \dots, p \end{aligned} \tag{15}$$

where $f(x)$, $g_i(x)$ and $h_j(x)$ are continuous $\forall i = 1, \dots, m, j = 1, \dots, p$. Let S denote the feasible set. The exterior penalty method, which is mainly useful for convex programs, solves (15) by a sequence of unconstrained minimization problems that converge to an optimal solution of (15) from outside of the feasible set. In the sequence, a penalty is imposed on every x outside the feasible set such that the penalty is increased from problem to problem, forcing the sequence to converge towards the feasible set.

To develop the algorithm, define the real-valued, continuous functions

$$\psi(\eta) = |\min(0, \eta)|^\alpha$$

and

$$\zeta(\eta) = |\eta|^\beta, \eta \in \mathbb{R}$$

where $\alpha, \beta \geq 1$ are given constants.

Let

$$s(x) = \sum_{i=1}^m \psi(g_i(x)) + \sum_{j=1}^p \zeta(h_j(x))$$

be the loss function for problem (15). Note that

$$\begin{aligned} s(x) &= 0 & \text{if } x \in S \\ s(x) &\geq 0 & \text{otherwise.} \end{aligned}$$

For any $p > 0$, we define the augmented objective function for problem (15) as

$$F(x, p) = f(x) + \frac{1}{p}s(x).$$

It is noted that $F(x, p) = f(x)$ if and only if x is feasible. Otherwise, $F(x, p) > f(x)$.

The exterior penalty method consists of solving a sequence of unconstrained optimizations for $k = 0, 1, 2, \dots$ given by

$$\min_x F(x, p^k) = f(x) + \frac{1}{p^k}s(x) \quad (16)$$

using a strictly increasing sequence of positive numbers, p^k . Let x^{k*} be the optimal solution to the k^{th} unconstrained optimization. The point x^{k*} is the initial point in the algorithm to solve (16). Then, the sequence of points $\{x^{k*}\}$, under mild conditions on (15), has a subsequence that converges to an optimal point of (15).

The exterior penalty method described is one of many that solves a sequence of unconstrained optimization problems to find the optimal solution of a non-linear programming problem. Another method that can be used to solve a non-linear programming problem, that doesn't require solving a sequence of optimization problems, is Lagrangian relaxation. This relaxation technique consists of embedding at least one of the constraints into the objective function with an associated Lagrangian multiplier μ , thus relaxing the constraint(s). The Lagrangian multipliers are simply the dual variables associated with the constraints. When a constraint is relaxed, it need not be satisfied, but a violation of the constraint penalizes the solution. The new, relaxed problem is then solved subject to the remaining constraints. Unlike the exact penalty method, the solution to a problem in which Lagrangian relaxation is applied is not an optimal solution of the original problem, but is still useful as bounds on the optimal solution can be determined.

The source for the material on Lagrangian relaxation is Ahuja et al.'s [11] and Conejo et al.'s [12].

Consider the Standard Simplex linear programming problem, (1)-(3).

We relax constraint (2). The new problem after applying Lagrangian relaxation is

$$\min \quad z = c^\top x + \mu^\top (Ax - b) \quad (17)$$

$$\text{s.t.} \quad x \geq 0. \quad (18)$$

In the problem given in (17)-(18), μ is the fixed vector of Lagrangian multipliers, which can be positive or negative, and has the same dimensions as the vector b . The Lagrangian multipliers are the dual variables associated with the constraints being relaxed. When $Ax \neq b$, x is not in the feasible region and the Lagrangian term in the objective function acts as a penalization. If the relaxed constraint is an inequality constraint, then the vector of Lagrangian multipliers would have to be all non-negative entries.

The function

$$L(\mu) = \min\{c^\top x + \mu(Ax - b) : x \geq 0\} \quad (19)$$

is referred to as the Lagrangian function.

Although the solution found from the relaxation is not optimal, a lower bound of the optimal value of (1) can be determined which gives useful information about the original problem.

LEMMA 1.26 (Lagrangian Bounding Principle). *For any vector μ of the Lagrangian multipliers, the value $L(\mu)$ of the Lagrangian function (19) is a lower bound on the optimal objective function value, z^* , of the original optimization problem given in (1)-(3).*

PROOF. Since $Ax = b$ for every feasible solution to the problem given in (1)-(3), for any vector μ of Lagrangian multipliers,

$$z^* = \min\{c^\top x : Ax = b, x \geq 0\} \quad (20)$$

$$= \min\{c^\top x + \mu(Ax - b) : Ax = b, x \geq 0\}. \quad (21)$$

Since removing the constraints $Ax = b$ from (21) cannot lead to an increase in the value of the objective function,

$$z^* \geq \min\{c^\top x + \mu^\top(Ax - b) : x \geq 0\} \iff z^* \geq L(\mu).$$

□

Since $L(\mu)$ is a lower bound on the optimal objective function value for any value of μ , the sharpest, or best, possible lower bound is found by solving the following optimization problem

$$L^* = \max_{\mu} L(\mu). \quad (22)$$

This optimization problem is referred to as the Lagrangian multiplier problem (also called the dual problem) associated with the problem given in (1)-(3). Applying the Weak Duality Theorem, we have that the optimal objective function value L^* of (22) is always a lower bound on the optimal value of (1). In other words, $L^* \leq z^*$. Then, for any Lagrangian multipliers μ , and any feasible solution x of the problem given in (1)-(3), we have

$$L(\mu) \leq L^* \leq z^* \leq c^\top x.$$

To find the optimal multiplier value μ^* of (22), we need to find the highest point on the Lagrangian function, (19). For convenience, Ahuja et al. refer to the quantity

$c^\top x + \mu^\top (Ax - b)$ as the composite cost of x . If the set $X = \{x^1, \dots, x^v\}$ is finite, then by definition,

$$L(\mu) \leq c^\top x^r + \mu^\top (Ax^r - b) \quad \forall r = 1, \dots, v.$$

In the space of composite costs and Lagrangian multipliers, μ , each function $y^r = c^\top x^r + \mu^\top (Ax^r - b)$ is a hyperplane. By definition of a hyperplane, its dimension is exactly one less than the dimension of the whole space. The Lagrangian function, (19), is the lower envelope of these hyperplanes for $r = 1, \dots, k$. We give a visual of this lower envelope in Figure 1.4. This visual is based on an example found in [11]. The optimization problem it is based on is not relevant to this section. We only include the graph of the composite costs to show how (19) is the lower envelope of the composite cost hyperplanes.

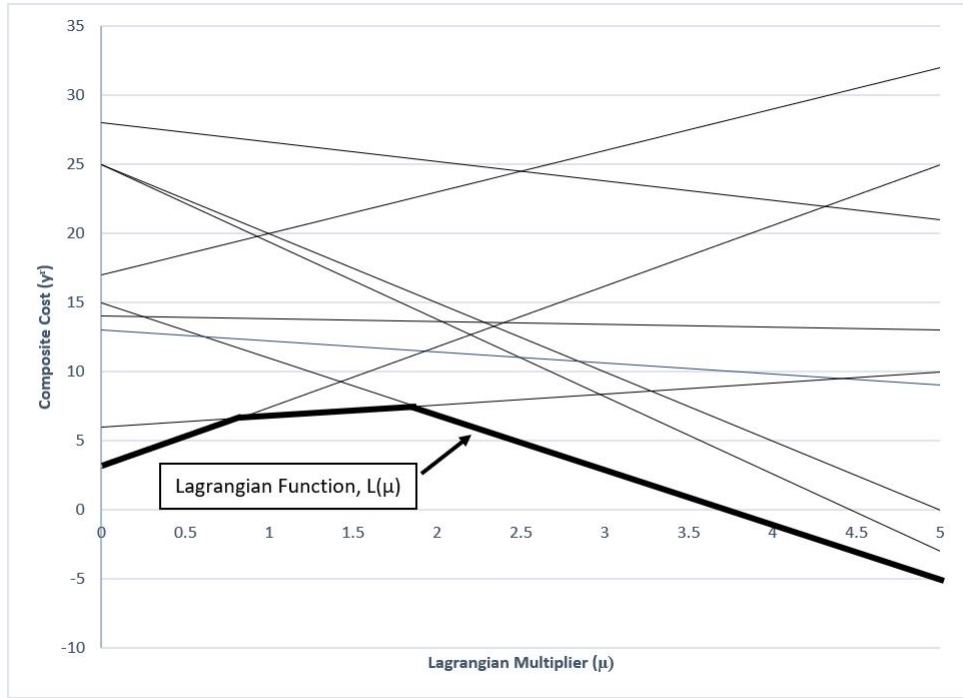


FIGURE 1.4. The Lagrangian Function

The bold, concave line in Figure 1.4 represents the Lagrangian function, (19). In the Lagrangian multiplier problem, (22), we wish to determine the highest point of (19). This point can be found by solving the Lagrangian master problem, given in (23)-(24).

$$\max \quad w \tag{23}$$

$$\text{s.t.} \quad w \leq c^\top x^r + \mu^\top (Ax^r - b) \quad \forall r = 1, \dots, v \tag{24}$$

This result is stated as a theorem.

THEOREM 1.27. *The Lagrangian multiplier problem $L^* = \max_{\mu} L(\mu)$ with $L(\mu) = \min\{c^\top x + \mu^\top (Ax - b) : x \geq 0\}$ is equivalent to the linear programming problem $L^* = \max\{w : w \leq c^\top x^r + \mu^\top (Ax^r - b) \forall r = 1, \dots, v\}$.*

1.5. Decomposition

The ideas in the section are based on the material by Conejo's et al. [12].

To begin the section, suppose an optimization problem has the structure shown in Figure 1.5.

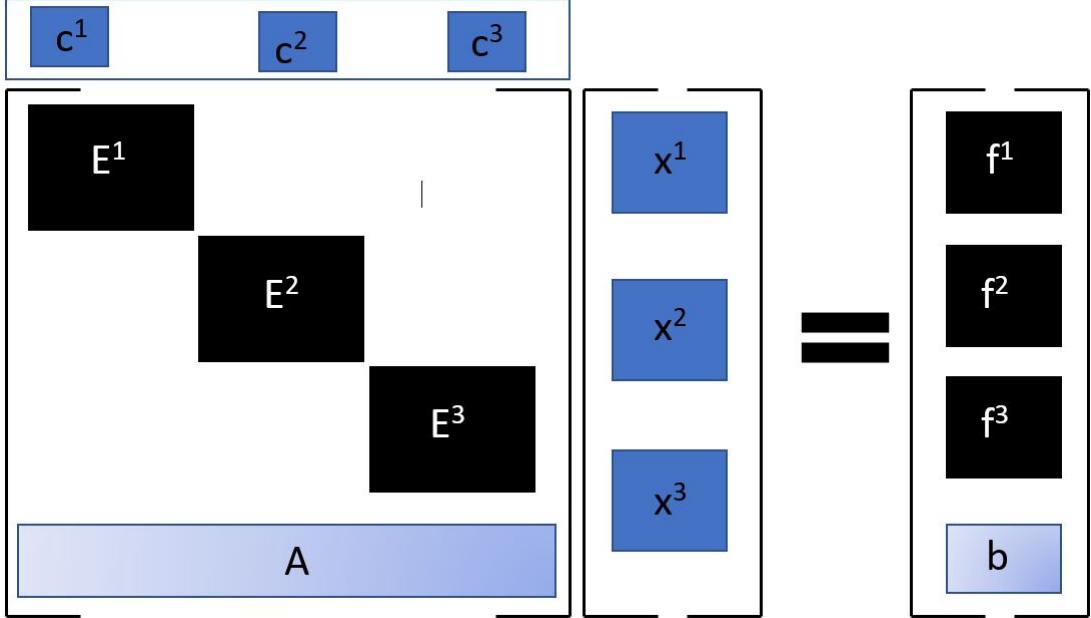


FIGURE 1.5. An Optimization Problem with Complicating Constraints

In Figure 1.5, the top rectangle with blocks c^1, c^2 and c^3 represents the objective function. The equation below the objective function is representative of the constraints of the optimization problem. If we ignore the block of constraints $Ax = b$, we see that the constraints have a staircase structure. That is, the problem can be decomposed into subproblems of the form

$$\begin{aligned} \min \quad & c^{k\top} x^k \\ \text{s.t.} \quad & E^k x^k = f^k \end{aligned}$$

for each $k = 1, 2, 3$.

This decomposable structure can not be achieved if we do not ignore the constraints $Ax = b$, which are called complicating constraints. In an optimization problem, constraints can be described as having decomposable structure, or being a complicating constraint. The advantage of decomposing a problem into several subproblems is a possible reduction in the solution time.

We now generalize this concept. Consider the following problem

$$\min \quad c^\top x \quad (25)$$

$$\text{s.t.} \quad Ex = f \quad (26)$$

$$Ax = b \quad (27)$$

$$x \geq 0. \quad (28)$$

In this problem, x is an $(n \times 1)$ vector. The matrices E and A are $(q \times n)$ and $(m \times n)$, respectively. Constraint (26) can be decomposed into r blocks, each of size $(q_k \times n_k)$, where $k = 1, \dots, r$. Constraint (27) is the complicating constraint. It does not have a decomposable structure. Constraint (28) sets a lower bound on the variables.

If the complicating constraint is ignored (relaxed), then the problem given in (25)-(28) becomes

$$\min \quad c^\top x \quad (29)$$

$$\text{s.t.} \quad Ex = f \quad (30)$$

$$x \geq 0. \quad (31)$$

The problem given in (29)-(31) can now be decomposed into k subproblems. The k^{th} subproblem is

$$\min \quad c^{[k]\top} x^{[k]} \quad (32)$$

$$\text{s.t.} \quad E^{[k]} x^{[k]} = f^{[k]} \quad (33)$$

$$x^{[k]} \geq 0. \quad (34)$$

EXAMPLE 1.28. The problem

$$\begin{array}{llll} \min & -10x_1 + 4x_3 - 7x_5 & & \\ \text{s.t.} & x_1 + x_2 & = & 10 \\ & & x_3 + x_4 & = 8 \\ & & & x_5 - x_6 = 1 \\ & x_1 + & x_3 - & x_6 = 11 \\ & x_i \geq 0 & \forall i = 1, \dots, 6 & \end{array}$$

has a decomposable structure into three blocks, where the last equality

$$x_1 + x_3 - x_6 = 11$$

is the complicating constraint.

The relaxed problem would be

$$\begin{array}{llll} \min & -10x_1 + 4x_3 - 7x_5 & & \\ \text{s.t.} & x_1 + x_2 & = & 10 \\ & & x_3 + x_4 & = 8 \\ & & & x_5 - x_6 = 1 \\ & x_i \geq 0 & \forall i = 1, \dots, 6 & . \end{array}$$

The three blocks would be:

$$\begin{aligned} \min \quad & -10x_1 \\ \text{s.t.} \quad & x_1 + x_2 = 10 \\ & x_1, x_2 \geq 0, \end{aligned}$$

$$\begin{aligned} \min \quad & 4x_3 \\ \text{s.t.} \quad & x_3 + x_4 = 8 \\ & x_3, x_4 \geq 0 \end{aligned}$$

and

$$\begin{aligned} \min \quad & -7x_5 \\ \text{s.t.} \quad & x_5 - x_6 = 1 \\ & x_5, x_6 \geq 0. \end{aligned}$$

Suppose each subproblem is solved p times with different, arbitrary objective functions. Assume that the p basic feasible solutions of the problem given in (29)-(31) are

$$X = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(p)} \\ x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(p)} \\ \vdots & \vdots & \ddots & \vdots \\ x_n^{(1)} & x_n^{(2)} & \cdots & x_n^{(p)} \end{bmatrix}.$$

where $x_j^{(s)}$ is the j^{th} component of solution s , $j = 1, \dots, n$ and $s = 1, \dots, p$.

The corresponding p optimal objective function values are

$$z = \begin{bmatrix} z^{(1)} \\ z^{(2)} \\ \vdots \\ z^{(p)} \end{bmatrix}$$

where $z^{(s)}$ is the objective function value of solution s .

The values of the m complicating constraints for the above p solutions are found by multiplying the matrix $A_{m \times n}$ by the matrix $X_{n \times p}$. The resulting m values are

$$R_{m \times p} = AX = \begin{bmatrix} r_1^{(1)} & r_1^{(2)} & \cdots & r_1^{(p)} \\ r_2^{(1)} & r_2^{(2)} & \cdots & r_2^{(p)} \\ \vdots & \vdots & \ddots & \vdots \\ r_m^{(1)} & r_m^{(2)} & \cdots & r_m^{(p)} \end{bmatrix}.$$

where $r_i^{(s)}$ is the value of the i^{th} complicating constraint for the s^{th} solution, $i = 1, \dots, m$.

EXAMPLE 1.29. Applying the above to Example 1.28, suppose we solve each subproblem $p = 2$ times, and we get the following solutions

$$x^{(1)} = \begin{bmatrix} 10 \\ 0 \\ 8 \\ 0 \\ 1 \\ 0 \end{bmatrix}, x^{(2)} = \begin{bmatrix} 10 \\ 0 \\ 0 \\ 8 \\ 0 \\ 1 \end{bmatrix}.$$

Then, we have

$$X = \begin{bmatrix} 10 & 10 \\ 0 & 0 \\ 8 & 8 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, z = \begin{bmatrix} -75 \\ -100 \end{bmatrix}.$$

The values of the complicating constraints are

$$R = \begin{bmatrix} 18 \\ 17 \end{bmatrix}.$$

To derive the master problem, we use the fact that a linear convex combination of basic feasible solutions of a linear program is a feasible solution of that linear program. This is true because the feasible region of a linear program is convex, which we showed in Example 1.13. The proof is trivial, so it is omitted.

We now solve the master (weighting) problem

$$\min \quad z^\top u \tag{35}$$

$$\text{s.t.} \quad Ru = b; \quad \lambda \tag{36}$$

$$e^\top u = 1; \quad \sigma \tag{37}$$

$$u \geq 0 \tag{38}$$

where u is a $(p \times 1)$ column vector of weights, e is a $(p \times 1)$ column vector of 1's, λ is the vector of dual variables corresponding to constraint (36) and σ is the vector of dual variables corresponding to constraint (37). The goal of the problem is to find the value of u which minimizes the value of all convex combinations of the p basic feasible solutions. Constraint (36) ensures that the complicating constraints of the original problem are enforced. Constraints (37) and (38) ensure that the weighting vector u satisfies the convex combination requirements.

A solution to the problem given in (35)-(38) is a convex combination of the p basic feasible solutions of (29)-(31). Thus, any solution to (35)-(38) is now a basic feasible solution to (29)-(31).

EXAMPLE 1.30. Continuing our example, the master problem would be

$$\begin{aligned} \min \quad & -75u_1 - 100u_2 \\ \text{s.t.} \quad & 18u_1 + 17u_2 = 11; \quad \lambda \\ & u_1 + u_2 = 1; \quad \sigma \\ & u_1, u_2 \geq 0. \end{aligned}$$

Consider a prospective new basic feasible solution is added to the problem given in (35)-(38) with objective function value z_j and complicating constraint values r_1, \dots, r_m .

The new master problem becomes

$$\min_{u, u_j} [z \quad z_j] [u \quad u_j]^\top \quad (39)$$

$$\text{s.t.} \quad [R \quad r_j] \begin{bmatrix} u \\ u_j \end{bmatrix} = b; \quad \lambda \quad (40)$$

$$e^\top \begin{bmatrix} u \\ u_j \end{bmatrix} = 1; \quad \sigma \quad (41)$$

$$\begin{bmatrix} u \\ u_j \end{bmatrix}^\top \geq 0 \quad (42)$$

where u_j is the weight corresponding to the prospective new basic feasible solution.

EXAMPLE 1.31. Now, let

$$x = \begin{bmatrix} 10 \\ 0 \\ 0 \\ 8 \\ 0 \\ 1 \end{bmatrix}$$

be a new prospective feasible solution.

The corresponding objective function value of this solution is $z = -107$, and its complicating constraint value is $r = 11$.

Then, our new master problem becomes

$$\begin{aligned} \min \quad & -75u_1 - 100u_2 - 107u_3 \\ \text{s.t.} \quad & 18u_1 + 17u_2 + 11u_3 = 11 \quad \lambda \\ & u_1 + u_2 + u_3 = 1; \quad \sigma \\ & u_1, u_2, u_3 \geq 0. \end{aligned}$$

If this new basic feasible solution is to be added to the set of previous ones, X , then the reduced cost of u_j should be negative and preferably a minimum.

The reduced cost of u_j can be computed as

$$\begin{aligned} d &= z_j - \lambda^\top r_j - \sigma \\ &= (c_j - A\lambda)^\top x_j - \sigma \end{aligned}$$

where x_j is the new prospective basic feasible solution. To find the minimum reduced cost, we solve

$$\min (c_j - A\lambda)^\top x_j \quad (43)$$

$$\text{s.t. } Ex_j = f \quad (44)$$

$$x_j \geq 0 \quad (45)$$

The σ in the objective function is omitted since it is a constant. Constraint (44) is included to make the new basic feasible solution feasible in the original problem, (25)-(28). The problem given in (43)-(45) can be solved in blocks, as it has the same structure as the relaxed problem, (29)-(31), just a different objective function.

EXAMPLE 1.32. The reduced cost associated with u_3 is

$$d = (1 - \lambda)x_1 + (1 - \lambda)x_3 - (1 + \lambda)x_6.$$

To find the minimum reduced cost, we solve

$$\begin{aligned} \min d \\ \text{s.t. } x_1 + x_2 &= 10 \\ x_3 + x_4 &= 8 \\ x_5 - x_6 &= 1 \\ x_i &\geq 0 \quad \forall i = 1, \dots, 6. \end{aligned}$$

This can be decomposed into three subproblems, similar to what we did in Example 1.28.

Using the solutions from each of the subproblems, we solve the optimal value of d , which we call v^0 .

Let

$$\nu^* = (c_j - A\lambda)^\top x_j^*$$

be the optimal value of [43]. Then,

$$d^* = \nu^* - \sigma^*$$

is the minimum reduced cost where σ^* is the optimal value of the dual variable associated with constraint (41).

If $\nu^* \geq \sigma^*$, the prospective basic feasible solution does not improve the reduced cost, so we do not include x_j in the set of previous solutions. If $\nu^* < \sigma^*$, the reduced cost is negative and can improve the objective function value, so we include

x_j in the set of previous basic feasible solutions.

EXAMPLE 1.33. To conclude our example, if the reduced cost is negative, the basic feasible solution x is added to the set of previous basic feasible solutions. So, our matrix X would now be

$$X = \begin{bmatrix} 10 & 10 & 10 \\ 0 & 0 & 0 \\ 8 & 8 & 0 \\ 0 & 0 & 8 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Otherwise, x is not added to the matrix, X .

The decomposition techniques described in this chapter are the techniques used in various solution algorithms. A well-known decomposition solution algorithm is the Dantzig-Wolfe decomposition algorithm. This algorithm uses the above decomposition concepts to find the solution to a linear program with complicating constraints. In this thesis, the problem is a concave minimization problem, thus the Dantzig-Wolfe decomposition technique cannot be used. Instead, Lagrangian relaxation is implemented as this technique can be used on non-linear programs. A big difference between the Dantzig-Wolfe and Lagrangian techniques is that the Dantzig-Wolfe master problem is in terms of primal variables, compared to the Lagrangian master problem which is in terms of dual variables.

CHAPTER 2

Green Supply Chain Network Design

2.1. Introduction

In this chapter, we present Elhedhli and Merrick's model and solution strategy.

Section 2.2 describes the problem formulation, and the assumptions that the model is based on. All variables and constants are defined, and the significance of the objective function and each of the constraints is described.

Section 2.3 outlines the strategies Elhedhli and Merrick used to find an optimal solution to the model introduced in Section 2.2. Lagrangian relaxation is used to relax the complicating constraint, thus allowing decomposition. A Lagrangian algorithm to find the best lower bound of the objective function from Section 2.2 is described. To find feasible solutions, a primal heuristic is then used. We show each step of the solution strategy.

In Section 2.4, we illustrate how Elhedhli and Merrick conducted their numerical testing. The way in which the constants were randomly generated is shown, as well as the way we retrieved the emissions data. The steps of the algorithm are also listed, which was implemented in MatLab and uses the optimization software, CPLEX, to solve the problem.

2.2. Elhedhli and Merrick's Model

This is a supply chain network design problem. The purpose of the problem is to find the optimal locations of the distribution centres (DCs), as well as the optimal assignments of plants to DCs and DCs to customers. This supply chain model is made up of three echelons, which are the levels in the supply chain: the plants, the DCs and the customers. Define the indices $i = 1, \dots, m$, $j = 1, \dots, n$ and $k=1, \dots, p$ which correspond to plant locations, potential distribution centers (DCs) and customers, respectively. A DC at location j has a maximum capacity of V_j and a fixed cost of g_j . Each customer has a demand of d_k . The variable cost of shipping a production unit from plant i to DC j is c_{ij} . Similarly, h_{jk} is the variable cost of shipping a production unit from DC j to customer k .

Let x_{ij} be the number of production units shipped from plant i to DC j . Let y_{jk} be 1 if customer k is assigned to DC j , and 0 otherwise. Let z_j be 1 if a DC is built at location j , and 0 otherwise.

Consider the mixed integer program

$$(FLM) \quad \min \quad \sum_{i=1}^m \sum_{j=1}^n f_{ij}(x_{ij}) + \sum_{j=1}^n \sum_{k=1}^p f_{jk}(d_k y_{jk}) + \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (46)$$

$$+ \sum_{j=1}^n \sum_{k=1}^p h_{jk} d_k y_{jk} + \sum_{j=1}^n g_j z_j$$

$$\text{s.t.} \quad \sum_{j=1}^n y_{jk} = 1, \quad \forall k, \quad (47)$$

$$\sum_{i=1}^m x_{ij} = \sum_{k=1}^p d_k y_{jk}, \quad \forall j, \quad (48)$$

$$\sum_{i=1}^m x_{ij} \leq V_j z_j, \quad \forall j, \quad (49)$$

$$\sum_{k=1}^p d_k y_{jk} \leq V_j z_j, \quad \forall j, \quad (50)$$

$$y_{jk}, z_j \in \{0, 1\}; x_{ij} \geq 0, \quad \forall i, j, k. \quad (51)$$

The terms $\sum_{i=1}^m \sum_{j=1}^n f_{ij}(x_{ij})$ and $\sum_{j=1}^n \sum_{k=1}^p f_{jk}(d_k y_{jk})$ of the objective function, (46), are measures of the carbon emissions from plant i to DC j and from DC j to customer k , respectively. The measure of carbon emissions is a function of the number of units being shipped and weight. This is because the number of units being shipped is directly proportional to the number of trucks needed for transportation. The subscripts on the functions correspond with where the trucks are travelling. The function $f_{ij}(\bullet)$ is the emissions cost of a truck travelling from plant i to DC j . Similarly, the function $f_{jk}(\bullet)$ is the emissions cost function of a truck travelling from DC j to customer k . Both emissions cost functions are concave functions.

The details of these functions are discussed in section 2.4.

The terms $\sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij}$ and $\sum_{j=1}^n \sum_{k=1}^p h_{jk}d_k y_{jk}$ of (46) are measures of the transportation cost from plant i to DC j and from DC j to customer k , respectively. The transportation cost is equal to the cost of shipping one unit multiplied by number of units being shipped. Both x_{ij} and $d_k y_{jk}$ are flow variables, and represent the number of units shipped in an echelon. The variable x_{ij} represents the number of units shipped from plant i to DC j . The variable $d_k y_{jk}$ represents the number of units shipped from DC j to customer k .

The term $\sum_{j=1}^n g_j z_j$ of (46) is a measure of the fixed cost of constructing DC j .

The objective function, (46), includes three costs: the emissions cost, transportation cost and fixed cost.

We now analyze the constraints. Constraint (47) ensures that each customer is assigned to only one DC. Constraint (48) ensures that the flow of goods into the DC is equal to the flow of goods out of the DC. This constraint links the echelons in the network. Together, constraints (47) and (48) ensure that total customer demand is satisfied. From constraint (47), each customer is assigned to a DC, and from constraint (48), the number of units coming into the DC is guaranteed to satisfy the demand of the customers assigned to that DC. Constraint (49) provides an upper bound to the number of units shipped to a DC j , that upper bound being the capacity of DC j . Constraint (50) provides an upper bound to the number of units shipped to all customers of a DC j , that upper bound being the capacity of DC j . These two constraints, (49) and (50), ensure that the number of units entering and leaving a DC j never exceeds that capacity of DC j . Constraint (51) sets y_{jk} and z_j as binary variables, and x_{ij} as a non-negative variable.

The Elhedhli and Merrick model is based on three assumptions. The first assumption is that the capacity of the plants is unlimited. This ensures that all customer demand can be met. It also allows for balancing at the DC (total input = total output). This assumption is feasible as a real world application because if a DC cannot have its capacity met by a supplier, they will hire an additional supplier, or change to a supplier who can meet their demand. The second assumption is that congestion or breakdowns of trucks are not taken into account, as it affects the environmental costing of the supply chain. Traffic congestion causes trucks to be on the road for a longer period of time, resulting in a higher emissions cost. Breakdowns of trucks also increases the amount of time trucks spend on the road, which results in a higher emissions cost. The third assumption is that total customer demand is satisfied. If total demand can not be satisfied, this implies that the DC does not have enough capacity to ship to customers.

2.3. Solution Strategy

Lagrangian relaxation is used on the problem given in (46)-(51) by relaxing constraint (48). This constraint links the echelons of the supply chain, and is a complicating constraint.

Recall the objective function (46). Relaxing the complicating constraint (48) using the fixed vector of Lagrangian multipliers μ_j , we obtain:

$$\begin{aligned}
& \sum_{i=1}^m \sum_{j=1}^n f_{ij}(x_{ij}) + \sum_{j=1}^n \sum_{k=1}^p f_{jk}(d_k y_{jk}) + \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n \sum_{k=1}^p h_{jk} d_k y_{jk} + \sum_{j=1}^n g_j z_j \\
& \quad - \sum_{j=1}^n \mu_j \left(\sum_{i=1}^m x_{ij} - \sum_{k=1}^p d_k y_{jk} \right) \\
\Longleftrightarrow & \sum_{i=1}^m \sum_{j=1}^n f_{ij}(x_{ij}) + \sum_{j=1}^n \sum_{k=1}^p f_{jk}(d_k y_{jk}) + \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n \sum_{k=1}^p h_{jk} d_k y_{jk} + \sum_{j=1}^n g_j z_j \\
& \quad - \sum_{j=1}^n \mu_j \sum_{i=1}^m x_{ij} + \sum_{j=1}^n \mu_j \sum_{k=1}^p d_k y_{jk} \\
\Longleftrightarrow & \sum_{i=1}^m \sum_{j=1}^n f_{ij}(x_{ij}) + \sum_{j=1}^n \sum_{k=1}^p f_{jk}(d_k y_{jk}) + \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n \sum_{k=1}^p h_{jk} d_k y_{jk} + \sum_{j=1}^n g_j z_j \\
& \quad - \sum_{i=1}^m \sum_{j=1}^n \mu_j x_{ij} + \sum_{j=1}^n \sum_{k=1}^p \mu_j d_k y_{jk} \\
\Longleftrightarrow & \sum_{i=1}^m \sum_{j=1}^n f_{ij}(x_{ij}) + \sum_{j=1}^n \sum_{k=1}^p f_{jk}(d_k y_{jk}) + \sum_{i=1}^m \sum_{j=1}^n (c_{ij} x_{ij} - \mu_j x_{ij}) \\
& \quad + \sum_{j=1}^n \sum_{k=1}^p (h_{jk} d_k y_{jk} + \mu_j d_k y_{jk}) + \sum_{j=1}^n g_j z_j \\
\Longleftrightarrow & \sum_{i=1}^m \sum_{j=1}^n f_{ij}(x_{ij}) + \sum_{j=1}^n \sum_{k=1}^p f_{jk}(d_k y_{jk}) + \sum_{i=1}^m \sum_{j=1}^n (c_{ij} - \mu_j) x_{ij} \\
& \quad + \sum_{j=1}^n \sum_{k=1}^p (h_{jk} d_k + d_k \mu_j) y_{jk} + \sum_{j=1}^n g_j z_j
\end{aligned}$$

The relaxed objective function penalizes the solution if constraint (48) is violated. In other words, if the difference between the number of units shipped to DC j and the demand of all of their customers is not 0, then the solution is penalized by a factor of μ_j .

The new mixed integer program can now be written as

$$(LR\text{-}FLM) \quad \min \sum_{i=1}^m \sum_{j=1}^n f_{ij}(x_{ij}) + \sum_{j=1}^n \sum_{k=1}^p f_{jk}(d_k y_{jk}) + \sum_{i=1}^m \sum_{j=1}^n (c_{ij} - \mu_j) x_{ij} \quad (52)$$

$$+ \sum_{j=1}^n \sum_{k=1}^p (h_{jk} d_k + d_k \mu_j) y_{jk} + \sum_{k=1}^p g_j z_j$$

$$\text{s.t.} \quad \sum_{j=1}^n y_{jk} = 1 \quad \forall k \quad (53)$$

$$\sum_{i=1}^m x_{ij} \leq V_j z_j \quad \forall j \quad (54)$$

$$\sum_{k=1}^p d_k y_{jk} \leq V_j z_j \quad \forall j \quad (55)$$

$$y_{jk}, z_j \in \{0, 1\}; x_{ij} \geq 0 \quad \forall i, j, k. \quad (56)$$

The purpose of applying Lagrangian relaxation to the original problem, (46)-(51), is to relax the complicating constraint. Now, the relaxed problem, given in (52)-(56), can be decomposed into subproblems. Specifically, it is decomposable by echelon.

The first subproblem is in terms of the binary variables y_{jk} and z_j . The solution of the problem determines the assignment of customers to DCs, and which DCs will open. We define (57)-(60) to be the first subproblem.

$$\min \quad \sum_{j=1}^n \sum_{k=1}^p f_{jk}(d_k y_{jk}) + \sum_{j=1}^n \sum_{k=1}^p (h_{jk} d_k + d_k \mu_j) y_{jk} + \sum_{k=1}^p g_j z_j \quad (57)$$

$$\text{s.t.} \quad \sum_{j=1}^n y_{jk} = 1 \quad \forall k \quad (58)$$

$$\sum_{k=1}^p d_k y_{jk} \leq V_j z_j \quad \forall j \quad (59)$$

$$y_{jk}, z_j \in \{0, 1\} \quad \forall i, j, k. \quad (60)$$

The first term in the objective function is the emissions cost, the second is the transportation and shipping cost from DC j to customer k , and the last term is the fixed cost of opening a DC at location j . We know that y_{jk} is binary, and $\sum_{j=1}^n y_{jk} = 1$. We also define $f_{jk}(0) = 0$. We can use these to simplify the objective

function of the above problem (57).

$$\begin{aligned}
& \sum_{j=1}^n \sum_{k=1}^p f_{jk}(d_k y_{jk}) + \sum_{j=1}^n \sum_{k=1}^p (h_{jk} d_k + d_k \mu_j) y_{jk} + \sum_{k=1}^p g_j z_j \\
\iff & \sum_{j=1}^n \sum_{k=1}^p f_{jk}(d_k) y_{jk} + \sum_{j=1}^n \sum_{k=1}^p (h_{jk} d_k + d_k \mu_j) y_{jk} + \sum_{k=1}^p g_j z_j \\
\iff & \sum_{j=1}^n \sum_{k=1}^p (f_{jk}(d_k) y_{jk} + (h_{jk} d_k + d_k \mu_j) y_{jk}) + \sum_{k=1}^p g_j z_j \\
\iff & \sum_{j=1}^n \sum_{k=1}^p (f_{jk}(d_k) + h_{jk} d_k + d_k \mu_j) y_{jk} + \sum_{k=1}^p g_j z_j \tag{61}
\end{aligned}$$

Now, we may re-define the first subproblem with its new objective function, (61)

$$(\text{SP1}) \quad \min \quad \sum_{j=1}^n \sum_{k=1}^p (f_{jk}(d_k) + h_{jk} d_k + d_k \mu_j) y_{jk} + \sum_{k=1}^p g_j z_j \tag{62}$$

$$\text{s.t.} \quad \sum_{j=1}^n y_{jk} = 1 \quad \forall k \tag{63}$$

$$\sum_{k=1}^p d_k y_{jk} \leq V_j z_j \quad \forall j \tag{64}$$

$$y_{jk}, z_j \in \{0, 1\} \quad \forall i, j, k. \tag{65}$$

The problem given in (62)-(65) is the first subproblem, and is a capacitated facility location problem with single sourcing.

The second subproblem is in terms of the flow variable x_{ij} , and its solution determines the number of units that will be shipped from each plant to a specific DC. We define (66)-(68) to be the second subproblem.

$$\min \quad \sum_{i=1}^m \sum_{j=1}^n f_{ij}(x_{ij}) + \sum_{i=1}^m \sum_{j=1}^n (c_{ij} - \mu_j) x_{ij} \tag{66}$$

$$\text{s.t.} \quad \sum_{i=1}^m x_{ij} \leq V_j z_j \quad \forall j \tag{67}$$

$$x_{ij} \geq 0 \quad \forall i, j. \tag{68}$$

If j is fixed, the above problem can be decomposed by potential warehouse site, resulting in n subproblems. One of the n subproblems is represented in (69)-(71).

$$\min \quad \sum_{i=1}^m f_{ij}(x_{ij}) + \sum_{i=1}^m (c_{ij} - \mu_j)x_{ij} \quad (69)$$

$$\text{s.t.} \quad \sum_{i=1}^m x_{ij} \leq V_j z_j \quad (70)$$

$$x_{ij} \geq 0 \quad \forall i. \quad (71)$$

When $z_j = 0$, a DC is not built at location j , making that case trivial. If that case is ignored, and we only consider the case when $z_j = 1$, then problem (69)-(71) becomes

$$\min \quad \sum_{i=1}^m f_{ij}(x_{ij}) + \sum_{i=1}^m (c_{ij} - \mu_j)x_{ij} \quad (72)$$

$$\text{s.t.} \quad \sum_{i=1}^m x_{ij} \leq V_j \quad (73)$$

$$x_{ij} \geq 0 \quad \forall i. \quad (74)$$

The objective function, (72), consists of the sum of concave functions, $f_{ij}(x_{ij})$, added to the sum of linear functions, $(c_{ij} - \mu_j)x_{ij}$. The sum of a concave functions is still concave, and a linear function is a concave function. So, (72) is concave.

Recall Corollary 1.25. This corollary is applied to the problem given in (72)-(74) to show that a global solution of this problem is achieved at some extreme point of its feasible domain. Let \mathbf{C} represent the feasible region of the problem given in (72)-(74).

In this problem, we know f , the emissions cost function, is concave. Recall that a line extends in both directions infinitely. Clearly, \mathbf{C} contains no lines since the constraints are bounded below by 0. Since \mathbf{C} is the feasible region of a linear programming problem, it follows from Example 1.13 that \mathbf{C} is a closed, convex set.

Since the problem given in (72)-(74) has an optimal solution at an extreme point of \mathbf{C} , we look closely at the structure of the extreme points. The extreme points of \mathbf{C} are at points where at most one x_{ij} takes on the value of V_j , and the remaining x_{ij} are equal to 0. This implies that the optimal solution will be at one of these points. This allows us to reformulate the problem given in (72)-(74) as

$$(\text{SP2j}) \quad \min \quad f_{ij}(V_j) + \sum_{i=1}^m (c_{ij} - \mu_j)x_{ij} \quad (75)$$

$$\text{s.t.} \quad \sum_{i=1}^m x_{ij} \leq V_j \quad (76)$$

$$x_{ij} \geq 0 \quad \forall i. \quad (77)$$

The relaxation of the original problem is now complete. The advantage of the relaxation is that now there are $n+1$ subproblems which can be solved with little

computational effort relative to the original problem. The n subproblems given in (75)-(77) can be solved relatively quickly. The first subproblem, given in (62)-(65), retains important characteristics of the initial problem such as the assignment of all customers to a single warehouse, and the condition that the demand of all customers is satisfied. These characteristics are retained in the feasible region of (62)-(65). A drawback of the relaxation is that (62)-(65) is now a capacitated facility location problem with single sourcing, which can be difficult to solve because the variables are binary. However, (62)-(65) is still easier to solve than the original problem, (46)-(51). By retaining the critical characteristics of the problem given in (46)-(51) in (62)-(65), a high quality Lagrangian bound can be achieved in a relatively small number of iterations. Further, using the solution of (62)-(65) in a primal heuristic, a high quality feasible solutions will be achieved.

The Lagrangian relaxation algorithm starts by initializing the Lagrangian multipliers and solving the subproblems. We denote the first subproblem, given in (62)-(65), as (SP1). We denote the second subproblem, given in (75)-(77), as (SP2j).

By Lemma 1.26, the lower bound of the relaxed problem given in (52)-(56) is defined as

$$LB = [\nu(SP1) + \sum_{j=1}^n \nu(SP2j)]$$

where $\nu(SP1)$ and $\nu(SP2j)$ are the values of the objective function at the optimal solutions to (SP1) and (SP2j), respectively.

The Lagrangian multiplier problem associated with the problem given in (52)-(56) is

$$LB^* = \max_{\mu} [\nu(SP1) + \sum_{j=1}^n \nu(SP2j)].$$

Let $h \in I_x$. Define I_x to be the index set of feasible integer points of the set

$$\left\{ (y_{jk}, z_j) : \sum_{j=1}^n y_{jk} = 1 \forall k; \sum_{k=1}^p d_k y_{jk} \leq V_j z_j \forall j; y_{jk}, z_j \in \{0, 1\}, \forall j, k \right\}.$$

For each possible feasible matrix $Y = [y_{jk}]$, the entries are binary and the sum of each column must be one. These matrices have dimension $(n \times p)$. Thus, at any given time, there are n choices of where the ‘1’ could be placed in each column, resulting in n^p possible feasible matrices, Y in the set. The vector $z = [z_j]$ can be determined through our choice of Y . Using the constraint

$$\sum_{k=1}^p d_k y_{jk} \leq V_j z_j$$

we find that

$$\begin{aligned} \sum_{k=1}^p y_{jk} = 0 &\implies z_j = 0 \\ \sum_{k=1}^p y_{jk} \neq 0 &\implies z_j = 1 \end{aligned}$$

Therefore, the number of possible pairs (Y, z) is n^p . The members of I_x index all of the possible pairs.

Let $h_j \in I_{y_j}$ Define I_{y_j} to be the index set of extreme points of the set

$$\left\{ (x_{ij}) : \sum_{i=1}^m x_{ij} \leq V_j; x_{ij} \geq 0, \forall i \right\}.$$

The number of extreme points, $x = [x_{ij}]$ in each of the j sets is $m + 1$, each of them being an $(m \times 1)$ vector. As mentioned earlier in the chapter, each extreme point of the set has the same structure: at most one of the x_{ij} will be equal to V_j , and the rest will be 0. Similarly, the members of I_{y_j} label each element of the j sets.

The best Lagrangian lower bound can be found by solving

$$\max_{\mu} \left\{ \min_{h \in I_x} \sum_{j=1}^n \sum_{k=1}^p (f_{jk}(d_k + h_{jk}d_k + d_k\mu_j)y_{jk}^h + \sum_{j=1}^n g_j z_j^h + \sum_{j=1}^n \min_{h_j \in I_{y_j}} f_{ij}(V_j) + \sum_{i=1}^m (c_{ij} - \mu_j)x_{ij}^{h_j}) \right\}. \quad (78)$$

Elhedhli and Merrick use Theorem 1.27 to reformulate (78) as the Lagrangian master problem, given in (79)-(81).

$$\begin{aligned}
& \max_{\theta_0, \theta_1, \dots, \theta_j, \mu_1, \dots, \mu_j} \theta_0 + \sum_{j=1}^n \theta_j \\
& \text{s.t.} \quad \theta_0 \leq \sum_{j=1}^n \sum_{k=1}^p (f_{jk}(d_k) + h_{jk}d_k + d_k\mu_j) y_{jk}^h + \sum_{j=1}^n g_j z_j^h \quad h \in I_x \\
& \quad \theta_j \leq \sum_{i=1}^m (f_{ij}(V_j)) + (c_{ij} - \mu_j) x_{ij}^{h_j} \quad h_j \in I_{yj}, \forall j
\end{aligned}$$

\Longleftrightarrow

$$\begin{aligned}
& \max_{\theta_0, \theta_1, \dots, \theta_j, \mu_1, \dots, \mu_j} \theta_0 + \sum_{j=1}^n \theta_j \\
& \text{s.t.} \quad \theta_0 \leq \sum_{j=1}^n \sum_{k=1}^p (d_k y_{jk}^h) \mu_j \\
& \quad + \sum_{j=1}^n \sum_{k=1}^p (f_{jk}(d_k) + h_{jk}d_k) y_{jk}^h \quad h \in I_x \\
& \quad \theta_j \leq \sum_{i=1}^m (f_{ij}(x_{ij}^{h_j})) + \sum_{i=1}^m c_{ij} x_{ij}^{h_j} - \sum_{i=1}^m \mu_j x_{ij}^{h_j} \quad h_j \in I_{yj}, \forall j
\end{aligned}$$

\Longleftrightarrow

$$\text{(LMP)} \quad \max_{\theta_0, \theta_1, \dots, \theta_j, \mu_1, \dots, \mu_j} \theta_0 + \sum_{j=1}^n \theta_j \tag{79}$$

$$\text{s.t.} \quad \theta_0 - \sum_{j=1}^n \left(\sum_{k=1}^p d_k y_{jk}^h \right) \mu_j \leq \sum_{j=1}^n \sum_{k=1}^p (f_{jk}(d_k) + h_{jk}d_k) y_{jk}^h \tag{80}$$

$$+ \sum_{j=1}^n g_j z_j^h \quad h \in I_x$$

$$\theta_j + \sum_{i=1}^m \mu_j x_{ij}^{h_j} \leq \sum_{i=1}^m (f_{ij}(x_{ij}^{h_j})) \tag{81}$$

$$+ \sum_{i=1}^m c_{ij} x_{ij}^{h_j} \quad h_j \in I_{yj}, \forall j$$

The Lagrangian master problem, (LMP), given in (79)-(81), can be solved as a linear programming problem. In (LMP), there are $n^p + n(m+1)$ constraints, which correspond to the number of feasible points that are indexed by I_x , and the extreme points that are indexed by each of the j sets, I_{yj} . As p increases, the number of constraints increases exponentially, which increases the amount of time needed to solve the problem. To shorten the computational time of solving (LMP)

we define a relaxation of the Lagrangian master problem using $\bar{I}_x \subset I_x$.

The second set of constraints, (81), contains redundant constraints. For the index set I_{yj} , the set of constraints associated with this set are of the form

$$\theta_j + \sum_{i=1}^m \mu_j x_{ij}^{h_j} \leq \sum_{i=1}^m \left(f_{ij} \left(x_{ij}^{h_j} \right) \right).$$

For each $h_j \in I_{yj}$, other than the index of the zero-vector, the left hand sides will be equivalent. The right hand sides will be the emissions cost of the capacity of DC j . So, the constraints will be

$$\theta_j + \sum_{i=1}^m \mu_j V_j \leq \sum_{i=1}^m \left(f_{ij} (V_j) \right).$$

Out of these m constraints, $m - 1$ will be redundant. The only significant constraint will be the one with the corresponding to the shortest distance between plant i and DC j , which in turn results in the lowest right hand side value. Therefore, considering all n index sets I_{yj} , out of the $n(m + 1)$ constraints, $n(m - 1)$ will be redundant. By including redundant constraints, we may be increasing the computational time needed to solve (79)-(81).

The relaxed formulation of (LMP), which we will refer to as (\overline{LMP}) , considers $h \in \bar{I}_x$. This relaxed master problem produces a new set of Lagrangian multipliers, and an upper bound to the full master problem, given in (79)-(81). Using these Lagrangian multipliers, we begin checking that constraints that we did not include in the relaxation are still satisfied. When we find a constraint from the index set I_x that isn't satisfied, we label it with the index \bar{a} , and a cut with the following form is generated:

$$\theta_0 - \sum_{j=1}^n \left(\sum_{k=1}^p d_k y_{jk}^{\bar{a}} \right) \mu_j \leq \sum_{j=1}^n \sum_{k=1}^p \left(f_{jk} (d_k) + h_{jk} d_k \right) y_{jk}^{\bar{a}} + \sum_{j=1}^n g_j z_j^{\bar{a}}$$

So, one cut is generated, the index set is updated as $\bar{I}_x = \bar{I}_x \cup \{\bar{a}\}$ and (\overline{LMP}) is solved with the new index set. The iteration ends when

$$|(\overline{LMP}) - (LB^*)| < \epsilon$$

The Lagrangian algorithm described provides the Lagrangian bound on the optimal solution. It does not, however, solve for the optimal combination of product flows, customer assignments and open facilities. A primal heuristic will be used, in addition to the Lagrangian algorithm, to generate feasible solutions.

Recall that (SP1) generates the assignments of customers to DCs, and determines whether a DC is opened or not at a location j . The demand of each DC, in units, can be determined using the two variables y_{jk}^h and z_j^h .

The problem given in (46)-(51) is reformulated as

$$\min \quad \sum_{j=1}^n \sum_{k=1}^p f_{jk} (d_k y_{jk}^h) + \sum_{j=1}^n \sum_{k=1}^p h_{jk} d_k y_{jk}^h + \sum_{i=1}^m \sum_{j=1}^n f_{ij} (x_{ij}) \quad (82)$$

$$+ \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n g_j z_j$$

$$\text{s.t.} \quad \sum_{i=1}^m x_{ij} = \sum_{k=1}^p d_k y_{jk}^h \quad \forall j \quad (83)$$

$$x_{ij} \geq 0 \quad \forall i, j. \quad (84)$$

The solutions y_{jk}^h and z_j^h are feasible in (SP1), thus we do not need to include the constraints of (SP1) in (82)-(84). We do not include the capacity constraint of warehouse j , since one of the constraints of (SP1) is

$$\sum_{k=1}^p d_k y_{jk}^h \leq V_j z_j, \forall j.$$

This, combined with the equality constraint, gives us

$$\sum_{i=1}^m x_{ij} \leq V_j z_j, \forall j$$

which ensures that the inflow into the warehouse does not exceed its capacity. The first two terms of (82) can be rewritten as one term, similar to the reformulation of (SP1) earlier in this chapter.

Then, the problem becomes

$$(\text{TP}) \quad \min \quad \sum_{j=1}^n \sum_{k=1}^p (f_{jk} (d_k) + h_{jk} d_k) y_{jk}^h + \sum_{i=1}^m \sum_{j=1}^n f_{ij} (x_{ij}) \quad (85)$$

$$+ \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n g_j z_j$$

$$\text{s.t.} \quad \sum_{i=1}^m x_{ij} = \sum_{k=1}^p d_k y_{jk}^h \quad \forall j \quad (86)$$

$$x_{ij} \geq 0 \quad \forall i, j. \quad (87)$$

This is a simple continuous flow transportation problem. In this transportation problem, we have the outflow from the supply node (DC) equal to the inflow into the demand node (customer). This network flow is continuous since our variable, x_{ij} , is a non-negative real number rather than an integer or a binary variable. The first and fourth terms of (85) are constants since we have values for y_{jk}^h and z_j^h . Thus, (85) can be rewritten as

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} (x_{ij}) + \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + C. \quad (88)$$

Since the first term of the new objective function, (88), is concave, and the second term is linear, which is also concave, then (88) is a concave function. By Corollary 1.25, (88) has an extreme point which is optimal. At an extreme point, all of the goods shipped to a DC comes from one plant, on a single truck. Since there is an extreme point which is optimal, each DC will be single-sourced by one plant, and the goods from that plant will be transported on a single truck. Thus, the optimal flow of units from plant to warehouse is equal to the DC demand or zero.

We reformulate the problem given in (85)-(87). We use constraint (86) to substitute

$$\sum_{i=1}^m x_{ij} \quad \text{for} \quad \sum_{k=1}^p d_k y_{jk}^h$$

in the reformulation.

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n f_{jk} \left(\sum_{k=1}^p d_k y_{jk}^h \right) w_{ij} + \sum_{i=1}^m \sum_{j=1}^n c_{ij} \left(\sum_{k=1}^p d_k y_{jk}^h \right) w_{ij} + C \\ \text{s.t.} \quad & \sum_{i=1}^m \left(\sum_{k=1}^p d_k y_{jk}^h \right) w_{ij} = \sum_{k=1}^p d_k y_{jk}^h & \forall j \\ & w_{ij} \in \{0, 1\} & \forall i, j \end{aligned}$$

\Longleftrightarrow

$$\text{(TP2)} \quad \min \quad \sum_{i=1}^m \sum_{j=1}^n \left(f_{jk} \left(\sum_{k=1}^p d_k y_{jk}^h \right) + c_{ij} \left(\sum_{k=1}^p d_k y_{jk}^h \right) \right) w_{ij} + C \quad (89)$$

$$\text{s.t.} \quad \sum_{i=1}^m \left(\sum_{k=1}^p d_k y_{jk}^h \right) w_{ij} = \sum_{k=1}^p d_k y_{jk}^h \quad \forall j \quad (90)$$

$$w_{ij} \in \{0, 1\} \quad \forall i, j \quad (91)$$

where

$$w_{ij} = \begin{cases} 1 & \text{if DC } j \text{ is supplied by plant } i; \\ 0 & \text{otherwise.} \end{cases}$$

In this reformulation, the optimal assignment of plant to DC is obtained. Constraint (90) ensures that the inflow into DC j is equal to the outflow, while simultaneously ensuring that one DC is assigned to only one plant. Constraint (91) defines w_{ij} to be a binary variable. There is no restriction on how many DCs can be assigned to a single plant, since the model is based on the assumption that the plants are uncapacitated. Thus, to find a solution, we look to minimize the cost of transportation, c_{ij} , over all i for each j .

EXAMPLE 2.1. Assume there are 2 plants, 3 warehouses and 5 customers. Let

$$Y = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

and

$$z = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

be the solution to (SP1).

Let

$$C = [c_{ij}] = \begin{bmatrix} 50 & 75 & 80 \\ 100 & 80 & 45 \end{bmatrix}$$

Then, by looking at the cost matrix, C , we choose the minimum value in each column to correspond to the assignment of plant to DC. In this case, c_{11} , c_{12} and c_{23} are the minimum costs in their respective columns, so the corresponding assignment matrix, W , would be

$$W = [w_{ij}] = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

This heuristic is activated at each iteration during numerical testing to find a feasible solution.

2.4. Numerical Testing

Elhedhli and Merrick coded the solution algorithm in MatLab 7, and within the code used CPLEX 11 to solve the subproblems, the heuristic and the master problems. They randomly generated problems, while keeping the parameters realistic. In our numerical testing of the algorithm, we attempted to replicate the algorithm proposed by Elhedhli and Merrick. Our solution algorithm was coded in MatLab 9.2 and uses CPLEX 12.9.

As in [1], the coordinates of the plants, distribution centers and customers were generated uniformly. To do so, two random numbers, a and b , were generated uniformly over $[10, 200]$. The pair (a, b) was then coordinate of the plant, DC or customer. From the coordinates, the Euclidean distance between each set of nodes was computed. From the Euclidean distance, we then set the transportation and handling costs between nodes.

$$\begin{aligned} c_{ij} &= \beta_1 \times (10 \times d_{ij}) \\ h_{jk} &= \beta_2 \times (10 \times d_{jk}) \end{aligned}$$

The parameters β_1 and β_2 were used by Elhedhli and Merrick in their numerical testing to test various scenarios. In our testing, we set $\beta_1 = \beta_2 = 1$.

The demand of each customer, d_k , was generated uniformly over $[10, 50]$.

The capacities of the DCs, V_j , were set to

$$V_j = \kappa \times (U[10, 160])$$

where κ was used to scale the ratio of warehouse capacity to demand. The scaling parameter κ dictates the rigidity of the problem. As κ increases, we give more choice as to where customers can receive their product. This has a large impact on the time required to solve the problem. The capacities, V_j , were scaled to satisfy

$$\kappa \in \{3, 5, 10\}.$$

The fixed cost of opening DC j was designed to reflect economies of scale. Economies of scale refers to the situation in which as the size of the DC being built increases, the marginal fixed cost of building a DC decreases. The marginal cost is the change in total cost when the size of the DC increases by one unit. The fixed cost of opening DC j was set to

$$g_j = \alpha \times \sqrt{V_j} \times (U[0, 90] + U[100, 110]).$$

The parameter α was used by Elhedhli and Merrick in their numerical testing to test various scenarios. In our testing, we set $\alpha = 1$.

Few data sets exist which show the relationship between vehicle weights and exhaust emissions. In Elhedhli and Merrick's paper, the emissions data is obtained using the US Environmental Protection Agency's (EPA's) Mobile6 computer program. This program contained an extensive database of CO₂ emissions for heavy-duty diesel vehicles of various weights. Since the publication of this paper, the EPA has updated this computer program. The latest version of this program is the MOtor Vehicle Emission Simulator (MOVES) 2014b. The emissions data is contained in MySQL Community version 5.7, and MOVES2014b uses this data to simulate emissions in different scenarios. One can choose the type of vehicle to consider, the US county to extract data from, the time of day to consider, as well as many other factors.

Instead of using the simulation, the emissions data was modeled based on Figure 2.1 in Elhedhli and Merrick's paper. The function corresponding to the graph was a function of vehicle weight in pounds which outputted carbon emissions in grams per kilometre traveled. In Elhedhli and Merrick's graph, four different speeds are shown. We modeled a speed of 100km/h and 60km/h, as they are the average highway speed and city speed of trucks, respectively.

The points (10 000, 380), (20 000, 510), (30 000, 630), (40 000, 700), (50 000, 750), (60 000, 770) were plotted in Excel to plot the emissions curve corresponding to a truck driving 100km/h. A logarithmic trendline was fit to the points, and the following function describing the data was obtained

$$e(x) = 228.52\ln(x) - 1732. \quad (92)$$

The emissions function, (92) is shown in Figure 2.1.

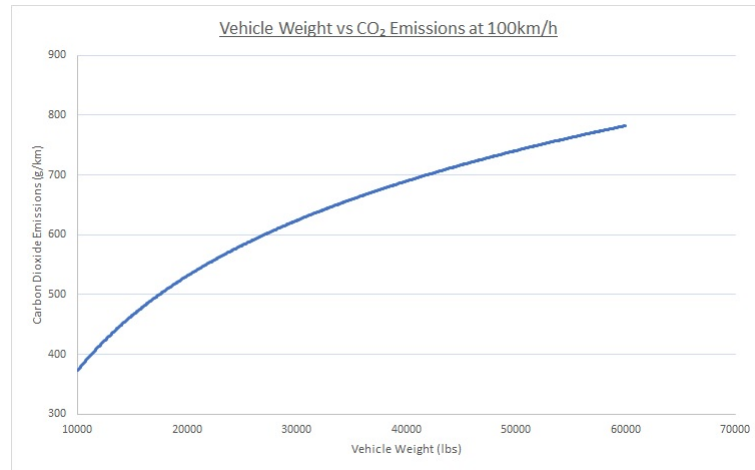


FIGURE 2.1. Vehicle weight vs. CO₂ emissions at 100km/h

The emissions cost function is considered in the test problem. To compute the emissions cost, the distance traveled, vehicle weight and emissions rate must be known. The vehicle weight is determined by number of units loaded onto the truck. This number is represented by either x_{ij} or $d_k y_{jk}$, depending on if the truck

is traveling to the DC or to the customer, respectively. An empty vehicle weight of 15 000lbs was assumed and the weight of a single production unit was assumed to be 75 lbs. It is assumed that single vehicle trips are made between nodes, so the emissions curve of a single truck is used in our problem. A travel speed of 100km/h was used to compute emissions levels, which is representative of average highway speed. The emissions cost of the network is determined by the equations

$$f_{ij}(15,000 + 75x_{ij}) = \Omega \times 0.2 \times e(15,000 + 75x_{ij}) \times d_{ij}$$

and

$$f_{jk}(15,000 + 75d_k y_{jk}) = \Omega \times 0.2 \times e(15,000 + 75d_k y_{jk}) \times d_{jk}.$$

Ω is a unitless scaling parameter, used to test various network parameters. The constant 0.2 is used for unit conversions and to associate a dollar value to the emission quantity. The units of this constant are dollars per gram, meaning that every gram of carbon emissions has an associated cost of 20 cents. The variables d_{ij} and d_{jk} represent the distance traveled from plant i to DC j , and from DC j to customer k , respectively. We include the subscripts on f to correspond to the distance traveled.

For simplicity, we write

$$f_{ij}(x_{ij}) = f(15,000 + 75x_{ij})$$

and

$$f_{jk}(d_k y_{jk}) = f(15,000 + 75d_k y_{jk})$$

We assume that empty vehicles will not be sent out from the plants or warehouses. So, in this thesis, $f_{ij}(0) = f_{jk}(0) = 0$.

We now outline the algorithm that was used to solve various scenarios.

[Step 0] of the algorithm is the initialization step. In this step, we randomly generate the coordinates of the plants, distribution centers and customers. We define the cost matrices C and H , then randomly generate the customer demand, DC capacity and fixed cost of opening a DC. Once these values are generated, we define the sets \bar{I}_x and I_{yj} . We want \bar{I}_x to index at least $n(p+1)+1$ feasible points, which is one more than the number of variables in (SP1). This ensures that we have enough constraints in (\overline{LMP}) to bound the problem. To determine which feasible points are indexed by \bar{I}_x , we randomly generate matrices that satisfy the constraints of (SP1). Each of the n sets, I_{yj} , indexes all of the extreme points of the n subproblems, (SP2j). The initialization step ends with the initialization of

the iteration counter, $\nu = 1$.

[Step 1] of the algorithm is solving (\overline{LMP}) using the function in CPLEX used to solve linear programming problems, 'cplexlp()'.

[Step 2] of the algorithm is solving (SP1) and (SP2j) using the new Lagrangian multipliers found in Step 1. The function in CPLEX used to solve binary integer programming problems, 'cplexbip()', is used to solve (SP1). The function in CPLEX used to solve mixed integer programming problems, 'cplexmilp()', is used to solve each subproblem, (SP2j). The objective function values of (SP1) and (SP2j) are then used to determine the best Lagrangian lower bound.

[Step 3] of the algorithm involves activating the heuristic using the solution (SP1) in Step 2 to obtain feasible solutions for subproblems (SP2j).

[Step 4] of the algorithm is the comparison of the objective function value of the Lagrangian master problem and the best Lagrangian lower bound.

If $|(\overline{LMP}) - (LB^*)| < \epsilon$

The algorithm terminates.

Else

Generate one cut.

Update the index set as $\bar{I}_x = \bar{I}_x \cup \{\bar{a}\}$.

Update the iteration counter, $\nu = \nu + 1$.

Go back to [Step 1].

CHAPTER 3

Results and Findings

3.1. Introduction

A relatively small problem consisting of three plants, seven DCs and fifteen customers is considered, and the algorithm described in Chapter 2 is implemented. To see how the inclusion of carbon emissions in a supply chain model affect its solution, we consider three cases in which the only changing variable is the parameter associated with the emissions cost function, Ω . For each case, we include a graph of the resulting network design. The first case we consider is when $\Omega = 0$, which describes a model that does not consider emissions costs. The second case we consider is when $\Omega = 1$, which described a model that considers moderate emissions costs. The third case we consider is when $\Omega = 5$, which described a model that considers high emissions costs. At the end of the chapter, we comment on how our results match those obtained by Elhedhli and Merrick.

3.2. Results and Findings

A relatively small problem consisting of three plants, seven DCs and fifteen customers is considered. We test three scenarios: one with zero emissions costs, one with moderate emissions costs and one with high emissions costs. The locations of the plants, DCs and customers are the same in all three cases. The locations of the potential DCs were randomly generated to have the following coordinates

$$\begin{bmatrix} 107 & 23 \\ 184 & 64 \\ 70 & 74 \\ 163 & 97 \\ 79 & 70 \\ 180 & 151 \\ 59 & 62 \end{bmatrix}$$

The fixed costs of opening a DC, the DC capacities and customer demands are also kept the same, and were randomly generated to be

$$g = [g_j] = \begin{bmatrix} 158.70 \\ 129.57 \\ 171.06 \\ 130.12 \\ 190.39 \\ 185.67 \\ 127.66 \end{bmatrix}, \kappa V = \kappa[V_j] = \begin{bmatrix} 471 \\ 423 \\ 291 \\ 147 \\ 207 \\ 387 \\ 312 \end{bmatrix}, d = [d_k] = \begin{bmatrix} 23 \\ 39 \\ 29 \\ 33 \\ 37 \\ 14 \\ 35 \\ 42 \\ 27 \\ 44 \\ 27 \\ 46 \\ 28 \\ 30 \\ 37 \end{bmatrix}$$

We consider a tight rigidity, $\kappa = 3$, for all three cases.

Figures 3.1, 3.2 and 3.3 are the resulting network designs of the three cases we consider. The plants in the network are denoted by circles, the DCs are denoted by triangles and the customers are denoted by squares. The different lines, dotted and solid, represent the shipping lanes of the different plant-DC-customer chains. The DC sites that are not used are crossed out with an “X”. The plants have unlimited capacity, so there is no limit on how many DCs each plant can service. In all three network designs, each DC is serviced by a single plant, and each customer is serviced by a single DC, which was required by the problem formulation. The total demand of all customers that must be satisfied is 491 units. Thus, the minimum number of DCs that must be opened to satisfy customer demand is two. For all three cases, the results we obtained had an error less than 5%.

To model the zero emissions cost case we set $\Omega = 0$, simply making the model a facility location problem. The resulting network design is shown in Figure 3.1. Recall that each variable, z_j , represents whether or not a DC is open at location j . The optimal z vector in this case was

$$z = [z_j] = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Four out of seven DCs were selected to open. The DC with the lowest capacity, and the two DCs with the highest fixed costs, were not selected to be open DC sites. Looking at the network design, there are two DCs open in the left half of the grid, and two DCs open in the right half. The DC with the highest capacity is servicing most of the customers in the left half of the grid. This solution contains two more than the minimum possible number of open DCs.

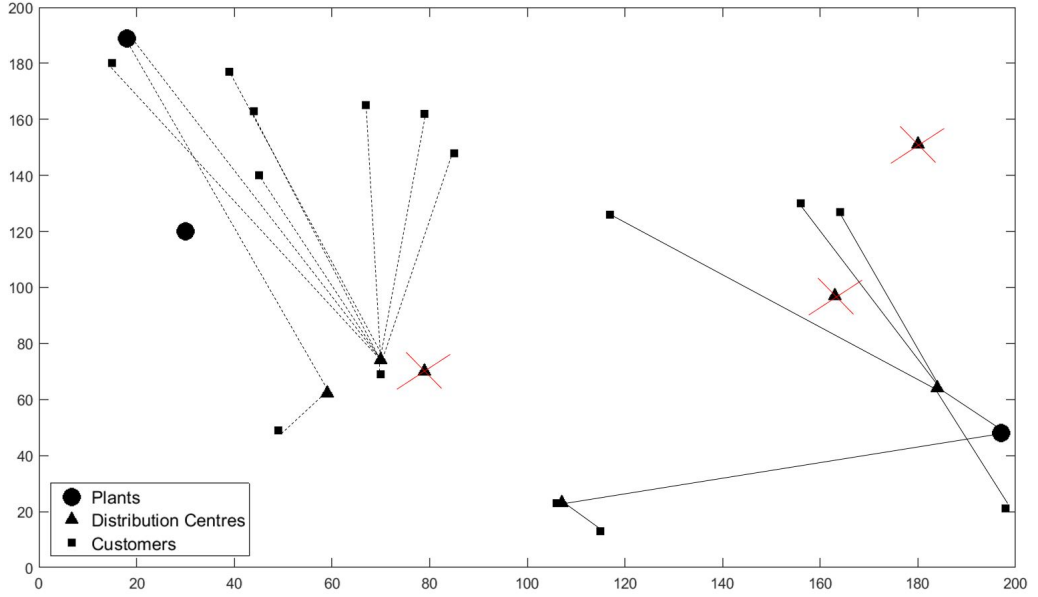


FIGURE 3.1. Network Design with Zero Emissions Costs

To model the moderate emissions cost case we set $\Omega = 1$. The resulting network design is shown in Figure 3.2. The optimal z vector in this case was

$$z = [z_j] = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Compared to the network design with zero emissions, one more DC was selected to open, resulting in five open DCs. The two DCs with the highest fixed costs were still not selected. The extra DC that was selected to open was open on the right half of the grid, and services three local customers. This reduces the distance traveled, which in turns reduces the emissions cost and the costs of transportation and shipping.

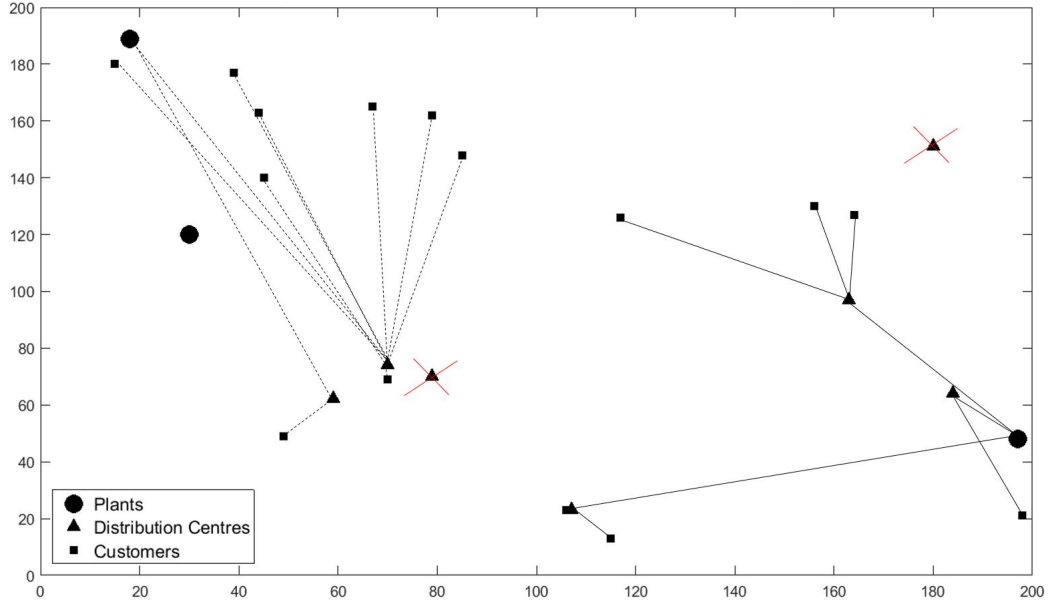


FIGURE 3.2. Network Design with Moderate Emissions Costs

To model the high emissions cost case we set $\Omega = 5$. The resulting network design is shown in Figure 3.3. The optimal z vector in this case was

$$z = [z_j] = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

The only DC which was not selected to open was the one with the highest fixed cost. Compared to the network design with zero emissions, DCs around the exterior of the grid are now open. This case produced the highest number of open DCs, which resulted in the network with the lowest vehicle kilometres traveled. This reduces the emissions costs and the cost of transportation and shipping more than the other two cases.

These results match the results of Elhedhli and Merrick. In the zero emissions case, only four DCs were selected to open, which were relatively central to where most of the customers were located. In the moderate and high emissions cases, it was fiscally feasible to open more DCs on the exterior of the grid to minimize the total

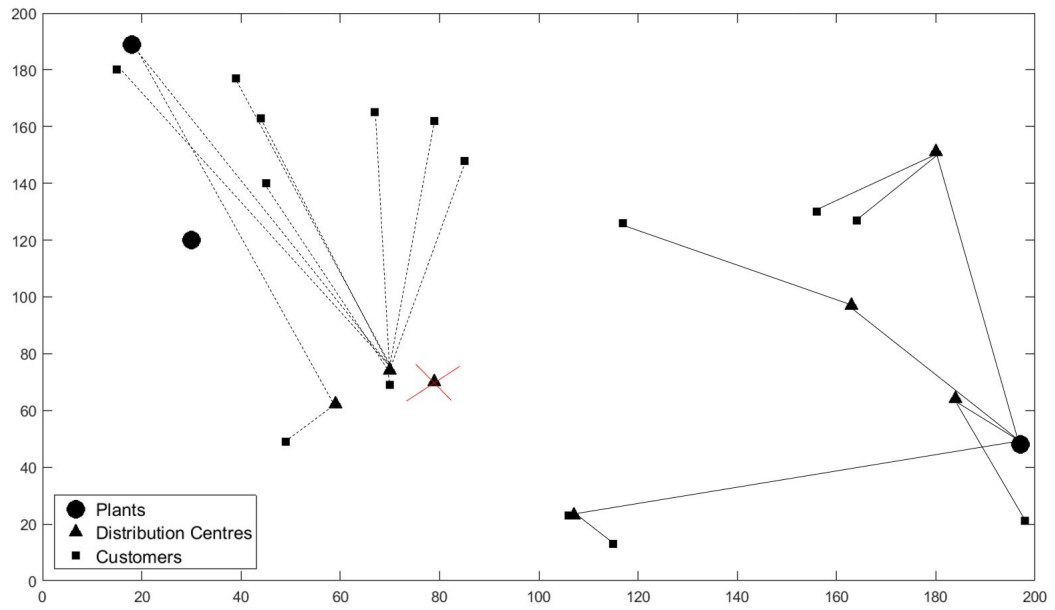


FIGURE 3.3. Network Design with High Emissions Costs

cost of the system. We have confirmed that the addition of carbon costs creates a pull to reduce the vehicle distance traveled. To do so, more DCs are selected to open to avoid long distances traveled from central DCs to outlier customers.

CHAPTER 4

Conclusions

4.1. Conclusion

This is an exploration of Elhedhli and Merrick's paper [1]. We gave an overview of topics that were used by Elhedhli and Merrick in the development of their green supply chain network design. These topics included: linear programming, convexity, Lagrangian relaxation and decomposition. Using these preliminary concepts, we demonstrated how Elhedhli and Merrick derived their model and solution strategy. Based on the solution strategy proposed by Elhedhli and Merrick, we constructed an algorithm which we used to solve a relatively small case study. The results of this case study confirmed the results published by Elhedhli and Merrick. The addition of carbon costs suggests that more DCs be opened to decrease the vehicle travel distances.

4.2. Summary of Contributions

The originality of this Master's thesis is in the organization and presentation. We reviewed several topics which aided in the understanding of how Elhedhli and Merrick's model was derived, why the solution strategy works, and how the solution strategy is implemented to achieve a high quality optimal solution. In particular, we studied the topics of linear programming, classic optimization problems, convexity, penalty methods, Lagrangian relaxation and decomposition. The two most significant topics studied were Lagrangian relaxation and decomposition. We compare Lagrangian relaxation to the exterior penalty method, and how they can be used to solve an optimization problem. We learned the theory behind Lagrangian relaxation, including how the Lagrangian function and the Lagrangian problem are related to the dual of an optimization problem. Further, we showed our understanding of how, when applying Lagrangian relaxation to an optimization problem, we can compute bounds on that problem. In terms of decomposition, we studied the notion of complicating constraints, and how relaxing constraints of this type can possibly reduce the solution time of an optimization problem. We introduce the concept of a master problem, and how dual variables can be used to determine the reduced cost of a solution. We relate the decomposition techniques we learned to a popular decomposition algorithm, which we then compare to the Lagrangian relaxation technique. When reviewing the contents and results of Elhedhli and Merrick's paper, we provide details which are not included in the paper. Based on the solution strategy proposed by Elhedhli and Merrick, we constructed an algorithm that iteratively solves the Lagrangian master problem by generating cuts, until the best Lagrangian lower bound is achieved. We used our algorithm on a relatively small case study, and our results confirmed the results published by Elhedhli and Merrick.

4.3. Future Work

This problem proposed by Elhedhli and Merrick is based on the assumption that full demand is always satisfied. Sometimes, it is more valuable for partial demand to be satisfied. For instance, if the location of a customer is an outlier, and it is expensive to serve that customer, then sending only a portion of the customer demand is more profitable. Allowing partial demand to be satisfied would change the model from minimizing cost to maximizing profit. A revenue for each product being shipped from DC to customer would have to be defined. Then, the objective function of the profit maximization problem would be the total revenue minus total costs, where the total costs are emissions cost, transportation cost and fixed cost of opening a DC.

Considering a multi-period model rather than a static one might be beneficial. A multi-period model is one that allows flexibility of parameters between different time periods. When costs, demands or other parameters change over time, one should test the value of a multi-period solution to see if it is worth considering. In Elhedhli and Merrick's model, the cost of transportation or emissions may change over time, as well as the demands of the customers or DCs.

Uncertainty is not considered in this model. There may be uncertainties in costs, demands and capacities. Specifically, uncertainty in capacities arise from risks such as environmental risks. By taking risk into consideration, the model goes from being deterministic to stochastic, which complicates the model. A deterministic model is one that will always have the same solution if the parameters stay the same. A stochastic model may or may not have the same solution, because probabilities are considered in the model. When including uncertainty in the model, it can make the model more complicated, thus making the method of solving the model more complicated as well. In the future, it may be valuable to consider a model with risk as it might make for a more accurate model.

Bibliography

- [1] Elhedhli, S., Merrick, R. (2012). Green supply chain network design to reduce carbon emissions. *Transportation Research Part D: Transport and Environment*, 17(5), 370-379. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1361920912000168>.
- [2] Environment Canada, 2019. *Greenhouse gas sources and sinks: executive summary 2019*. Retrieved from <https://www.canada.ca/en/environment-climate-change/services/climate-change/greenhouse-gas-emissions/sources-sinks-executive-summary-2019.html>.
- [3] Pardalos, P. & Rosen, J. (1986). Methods for global concave minimization: a bibliographic survey. *SIAM Review*, 28(3), 367-379. Retrieved from <https://www.jstor.org/stable/2031354>.
- [4] Best, M.J., & Ritter K. (1985). *Linear Programming: active set analysis and computer programs*. Englewood Cliffs: Prentice Hall, NJ.
- [5] Whittle, P. (2007). *Networks optimisation and evolution*. Cambridge University Press, Cambridge.
- [6] Hillier, Lieberman (2001). *Introduction to Operations Research, Seventh Edition*. McGraw-Hill, New York, NY.
- [7] Conforti, M., Cornuéjols, G., & Zambelli, G. (2014). *Integer Programming*. Springer International Publishing.
- [8] R.T.Rockafellar (1970). *Convex Analysis*. Princeton Univ. Press, Princeton, NJ.
- [9] Avriel, M (1976). *Nonlinear Programming: Analysis and Methods*. Englewood Cliffs, Prentice Hall, NJ.
- [10] Zangwill, W. I. (1967). Non-Linear Programming Via Penalty Functions. *Management Science*, 13(5), 344-358. Retrieved from <https://www.jstor.org/stable/2627851>.
- [11] R. K. Ahuja, T. L. Magnanti and J. B. Orlin (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.

- [12] A.J Conejo, E. Castillo, R. Minguez and R. Garcia-Bertrand (2011). *Decomposition Techniques in Mathematical Programming: Engineering and Science Applications*. Springer, Berlin.

Appendix A

Problem Reformulations for CPLEX Compatibility

As mentioned in section 2.4, CPLEX v12.9 was implemented into MatLab. By implementing CPLEX into MatLab, a toolbox of functions that are able to solve a variety of programming problems became available. Specifically, we used the linear programming problem solver function, as well as the binary linear programming solver function. The only problems we used a CPLEX function to solve were the first subproblem (SP1) and the relaxed Lagrangian master problem (\overline{LMP}). To use these two functions, our input variables had to coincide with the general linear programming problem form:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & A_1 x \leq b_1 \\ & A_2 x = b_2 \end{aligned}$$

where c, b_1 and b_2 are vectors, A_1 and A_2 are matrices, and x is the vector of variables we are minimizing over. We had to reformulate the (SP1) and (\overline{LMP}) to fit this form.

We will first show the reformulation of (SP1). The first step to the reformulation was to represent the problem in matrix form, rather than by summations. Starting with the objective function, we split it into three parts and reformulate them separately. Recall that the objective function of (SP1) is

$$\sum_{j=1}^n \sum_{k=1}^p (f_{jk}(d_k) + h_{jk}d_k + d_k\mu_j)y_{jk} + \sum_{k=1}^p g_k z_k.$$

We first reformulate the term

$$\sum_{j=1}^n \sum_{k=1}^p f_{jk}(d_k)y_{jk}. \tag{93}$$

Define

$$d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_p \end{bmatrix}, F_{jk} = \begin{bmatrix} f_{11}(d_1) & f_{12}(d_2) & \cdots & f_{1p}(d_p) \\ f_{21}(d_1) & f_{22}(d_2) & \cdots & f_{2p}(d_p) \\ \vdots & \ddots & & \vdots \\ f_{n1}(d_1) & f_{n2}(d_2) & \cdots & f_{np}(d_p) \end{bmatrix} \text{ and } Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1p} \\ y_{21} & y_{22} & \cdots & y_{2p} \\ \vdots & \ddots & & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{np} \end{bmatrix}$$

We define the operation \cdot^* to be element-wise matrix multiplication.

EXAMPLE 4.1. Let

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } B = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

then

$$A.*B = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$$

In (93), each element of the matrices F_{jk} and Y are being multiplied element wise, then added to each other. Thus, we can reformulate (93) to be

$$e_n^\top (F_{jk}.*Y)e_p$$

where e_n and e_p are a $(n \times 1)$ and a $(p \times 1)$ vector of ones, respectively.

We now reformulate the term

$$\sum_{j=1}^n \sum_{k=1}^p (h_{jk}d_k + d_k\mu_j)y_{jk}.$$

We can rewrite the above term as

$$\sum_{j=1}^n \sum_{k=1}^p d_k (h_{jk}y_{jk} + \mu_j y_{jk}). \quad (94)$$

Define

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} \text{ and } H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1p} \\ h_{21} & h_{22} & \cdots & h_{2p} \\ \vdots & \ddots & & \vdots \\ h_{n1} & h_{n2} & \cdots & h_{np} \end{bmatrix}$$

We use $[A]_{r\bullet}$ to denote the r^{th} row of a matrix A and $[A]_{\bullet c}$ to denote the c^{th} column of a matrix A .

As the reformulation of (94) is complicated, we show a small example to get a sense of how we can change this term.

EXAMPLE 4.2. Let $n = 2$. When we fix the column to $k = 1$, we get

$$\begin{aligned} \sum_{j=1}^2 d_1 (h_{j1}y_{j1} + \mu_j y_{j1}) &= d_1 (h_{11}y_{11} + \mu_1 y_{11}) + d_1 (h_{21}y_{21} + \mu_2 y_{21}) \\ &= d_1 (h_{11}y_{11} + h_{21}y_{21} + \mu_1 y_{11} + \mu_2 y_{21}) \\ &= d_1 \left(\begin{bmatrix} h_{11} & h_{21} \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{21} \end{bmatrix} + \begin{bmatrix} \mu_1 & \mu_2 \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{21} \end{bmatrix} \right). \end{aligned}$$

Generalizing Example 4.2, we may reformulate (94) as

$$d^\top T$$

where

$$T = \begin{bmatrix} [H]_{\bullet 1}^\top [Y]_{\bullet 1} + \mu^\top [Y]_{\bullet 1} \\ \vdots \\ [H]_{\bullet p}^\top [Y]_{\bullet p} + \mu^\top [Y]_{\bullet p} \end{bmatrix}.$$

The last term we need to reformulate is

$$\sum_{j=1}^p g_j z_j \tag{95}$$

which is simply an inner product.

Define

$$g = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix} \text{ and } z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}.$$

Then, the reformulation of (95) is

$$g^\top z.$$

Putting it all together, the objective function value of problem (SP1) in matrix form is now

$$e_n^\top (F_{jk} \cdot Y) e_p + d^\top T + g^\top z \tag{96}$$

To make the (96) compatible with the CPLEX functions, the objective function must be in the form

$$c^\top x$$

where x is a vector of variables. Again, we use a small example to see how we can get (96) to fit this form.

EXAMPLE 4.3. Let $n = 2$ and $p = 3$.

Then,

$$\begin{aligned}
e_n^\top (F_{jk} \cdot Y) e_p &= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} f_{11}(d_1)y_{11} & f_{12}(d_2)y_{12} & f_{13}(d_3)y_{13} \\ f_{21}(d_1)y_{21} & f_{22}(d_2)y_{22} & f_{23}(d_3)y_{23} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} f_{11}(d_1)y_{11} + f_{21}(d_1)y_{21}, & f_{12}(d_2)y_{12} + f_{22}(d_2)y_{22}, & f_{13}(d_3)y_{13} + f_{23}(d_3)y_{23} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \\
&= f_{11}(d_1)y_{11} + f_{21}(d_1)y_{21} + f_{12}(d_2)y_{12} + f_{22}(d_2)y_{22} + f_{13}(d_3)y_{13} + f_{23}(d_3)y_{23} \\
&= \begin{bmatrix} f_{11}(d_1), & f_{21}(d_1), & f_{12}(d_2), & f_{22}(d_2), & f_{13}(d_3), & f_{23}(d_3) \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{21} \\ y_{12} \\ y_{22} \\ y_{13} \\ y_{23} \end{bmatrix}
\end{aligned}$$

Then, generalizing the results from Example 4.3, and easily reformulating $d^\top T$, we have the following

$$\begin{bmatrix} [F_{jk}]_{\bullet 1}^\top + ([H]_{\bullet 1} + \mu_1)^\top, & [F_{jk}]_{\bullet 2}^\top + ([H]_{\bullet 2} + \mu_2)^\top, & \cdots & [F_{jk}]_{\bullet p}^\top + ([H]_{\bullet p} + \mu_p)^\top \end{bmatrix} \begin{bmatrix} [Y]_{\bullet 1} \\ [Y]_{\bullet 2} \\ \vdots \\ [Y]_{\bullet p} \end{bmatrix}.$$

The final step is to add the term $g^\top z$, so that the reformulation of the objective function from (SP1) becomes

$$\begin{bmatrix} [F_{jk}]_{\bullet 1}^\top + ([H]_{\bullet 1} + \mu_1)^\top, & [F_{jk}]_{\bullet 2}^\top + ([H]_{\bullet 2} + \mu_2)^\top, & \cdots & [F_{jk}]_{\bullet p}^\top + ([H]_{\bullet p} + \mu_p)^\top, & g^\top \end{bmatrix} \begin{bmatrix} [Y]_{\bullet 1} \\ [Y]_{\bullet 2} \\ \vdots \\ [Y]_{\bullet p} \\ z \end{bmatrix}. \quad (97)$$

We now reformulate the constraints. First, we look at the equality constraint, which must be in the form $Ax_1 = b_1$.

Recall the equality constraint from problem (SP1)

$$\sum_{j=1}^n y_{jk} = 1 \quad \forall k.$$

In other words, the sum of each column of matrix Y is 1. We may represent this constraint in the following way, which is compatible with the CPLEX functions

$$\begin{bmatrix} e_n^\top & 0_n^\top & 0_n^\top & \cdots & 0_n^\top & 0_n^\top \\ 0_n^\top & e_n^\top & 0_n^\top & \cdots & 0_n^\top & 0_n^\top \\ \vdots & \ddots & & \cdots & & \vdots \\ 0_n^\top & 0_n^\top & 0_n^\top & \cdots & e_n^\top & 0_n^\top \end{bmatrix} \begin{bmatrix} [Y]_{\bullet 1} \\ [Y]_{\bullet 2} \\ \vdots \\ [Y]_{\bullet p} \\ z \end{bmatrix} = [e_p] \quad (98)$$

where 0_n is an $(n \times 1)$ zero vector. The coefficient matrix of (98) has dimension $(p \times n(p+1))$.

The inequality constraint from (SP1) is

$$\sum_{k=1}^p d_k y_{jk} \leq V_j z_j.$$

This constraint can be represented as

$$\begin{bmatrix} d_1 I_n & d_2 I_n & \cdots & d_p I_n & -VI \end{bmatrix} \begin{bmatrix} [Y]_{\bullet 1} \\ [Y]_{\bullet 2} \\ \vdots \\ [Y]_{\bullet p} \\ z \end{bmatrix} \leq 0_n \quad (99)$$

where I_n is the $(n \times n)$ identity matrix, and

$$-VI = \begin{bmatrix} -V_1 & 0 & 0 & \cdots & 0 \\ 0 & -V_2 & 0 & \cdots & 0 \\ \vdots & \ddots & & \cdots & -V_n \end{bmatrix}.$$

The coefficient matrix of (99) has dimension $(p \times n(p+1))$.

The reformulations done for (SP2j) and the Lagrangian master problem are very similar to what we did above, so the descriptions are omitted.

Vita Auctoris

Penelopi Tatiana Krikella was born in 1996 in Windsor, Ontario. She graduated from St. Joseph's Catholic High School in 2014. She then obtained her B.Math degree in Mathematics from the University of Windsor in 2018. She is currently a candidate for an M.Sc degree in Mathematics at the University of Windsor, and hope to complete her degree by the end of 2019. Her research interests are in supply chain optimization.